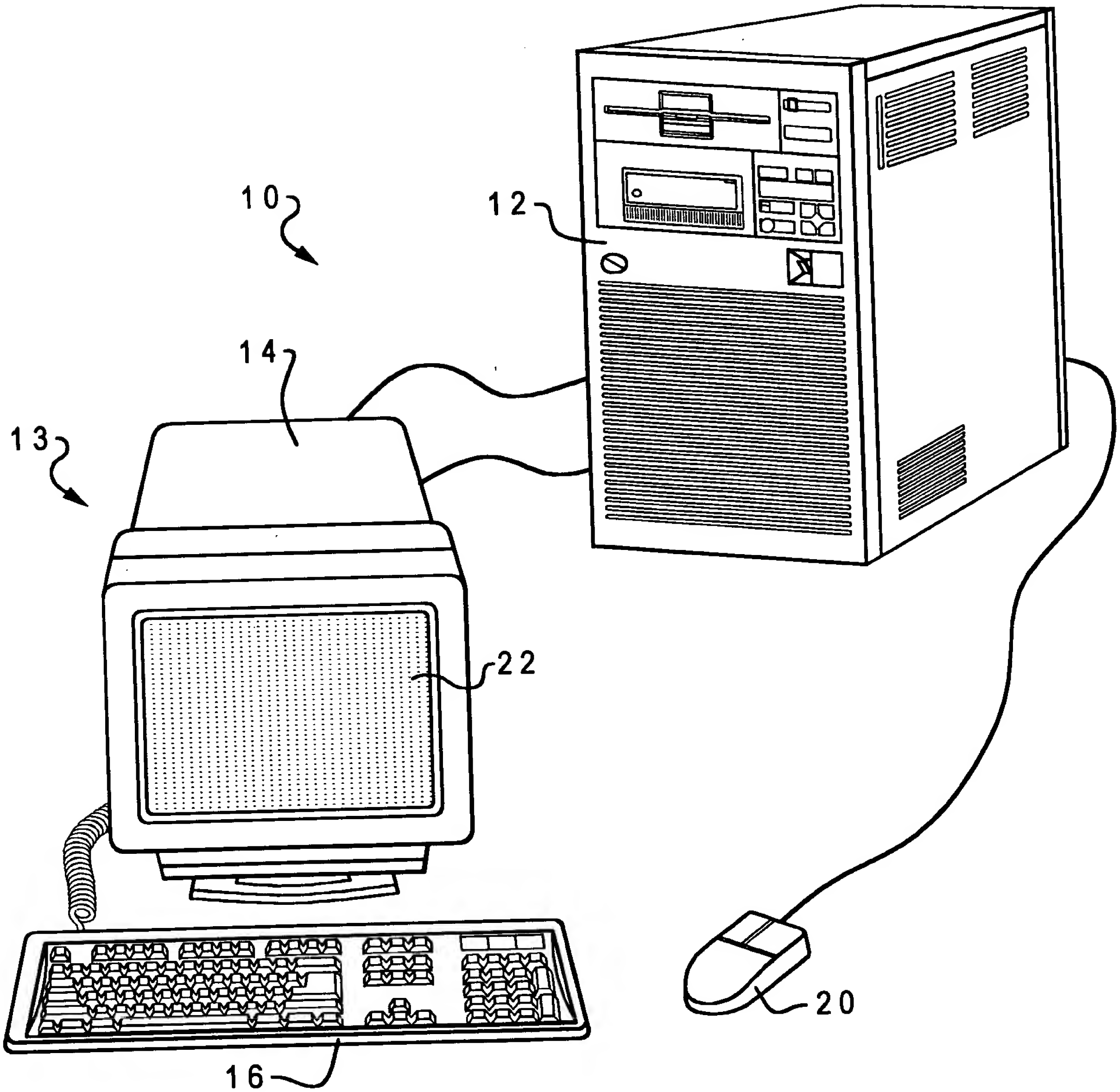


Fig. 1

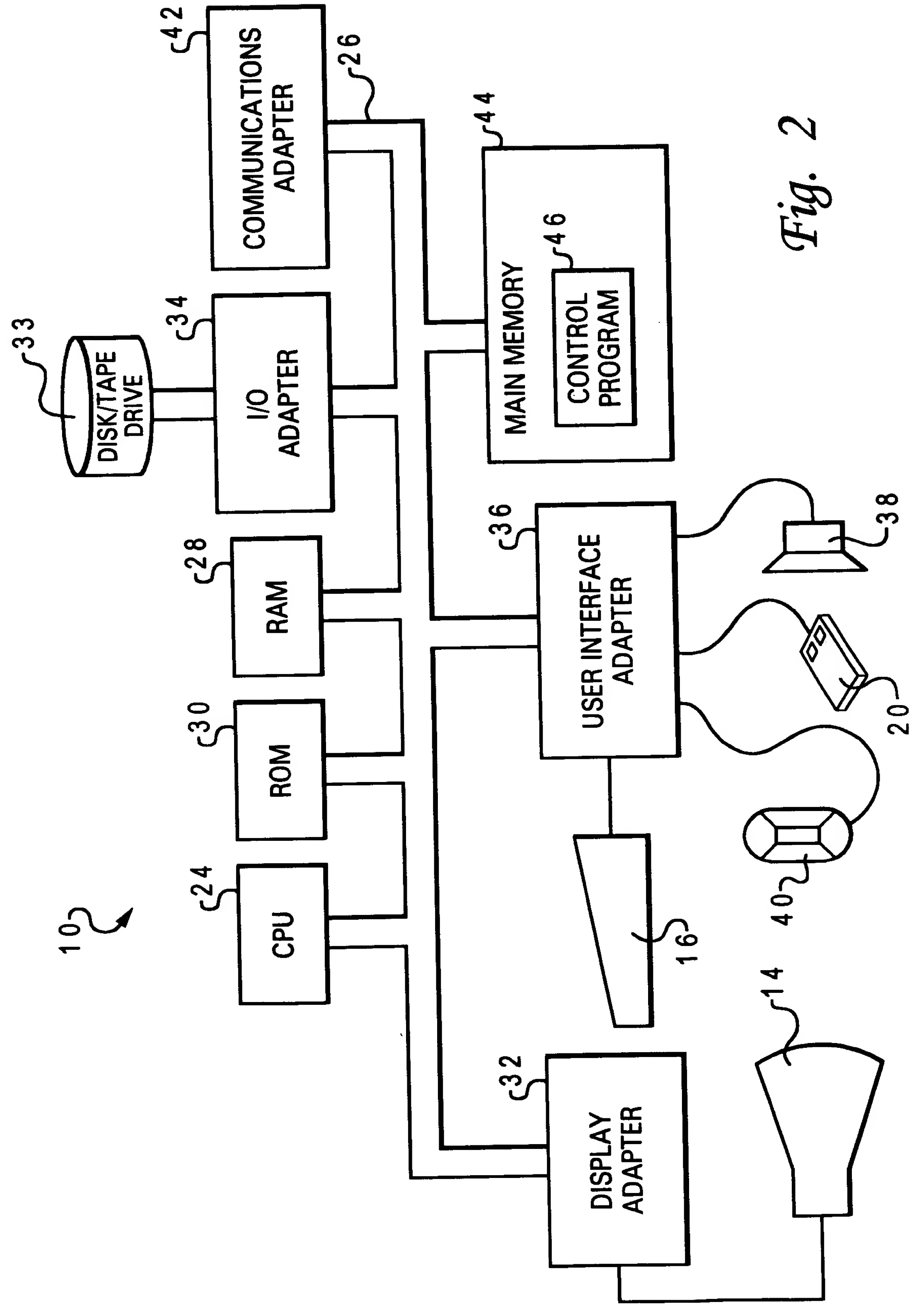

10
CPU
24
ROM
30
RAM
28
I/O ADAPTER
34
DISK/TAPE DRIVE
33
COMMUNICATIONS ADAPTER
42
26
MAIN MEMORY
44
CONTROL PROGRAM
46
USER INTERFACE ADAPTER
36
38
20
40
16
DISPLAY ADAPTER
32
14

Fig. 2

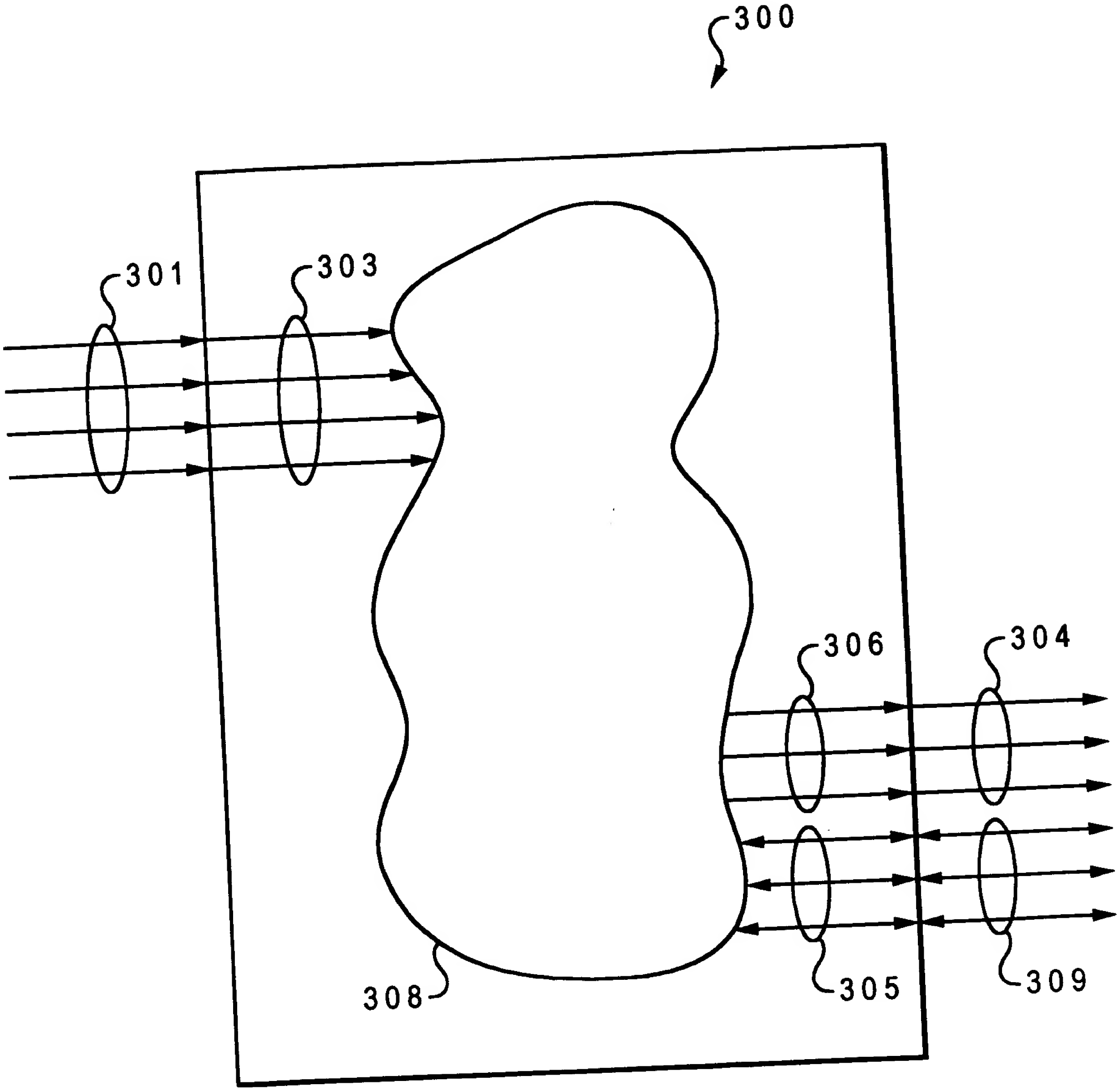


Fig. 3A

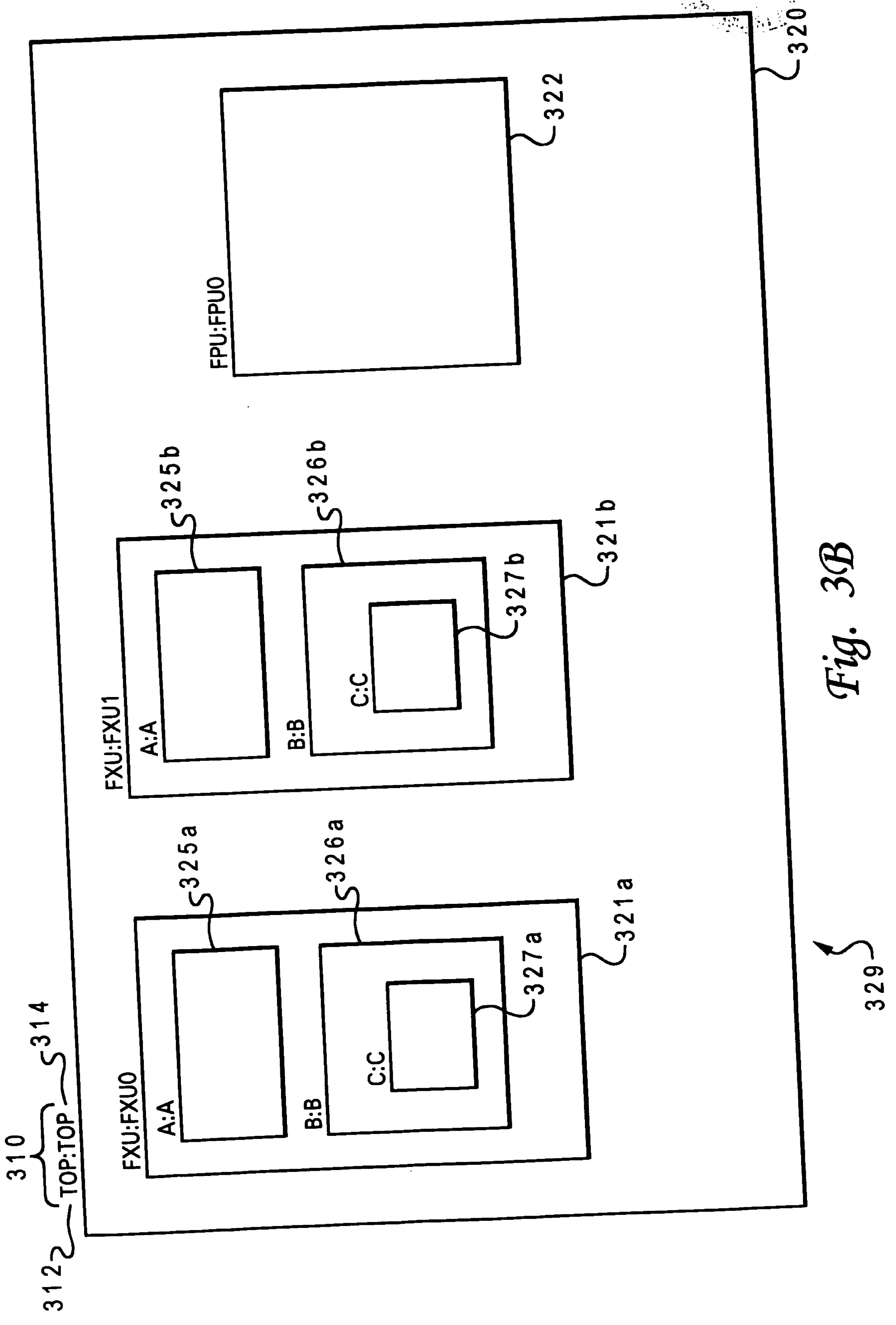

TOP:TOP
310
312
314
320
FXU:FXU0
A:A
B:B
C:C
321a
325a
326a
327a
FXU:FXU1
A:A
B:B
C:C
321b
325b
326b
327b
FPU:FPU0
322
329

Fig. 3B

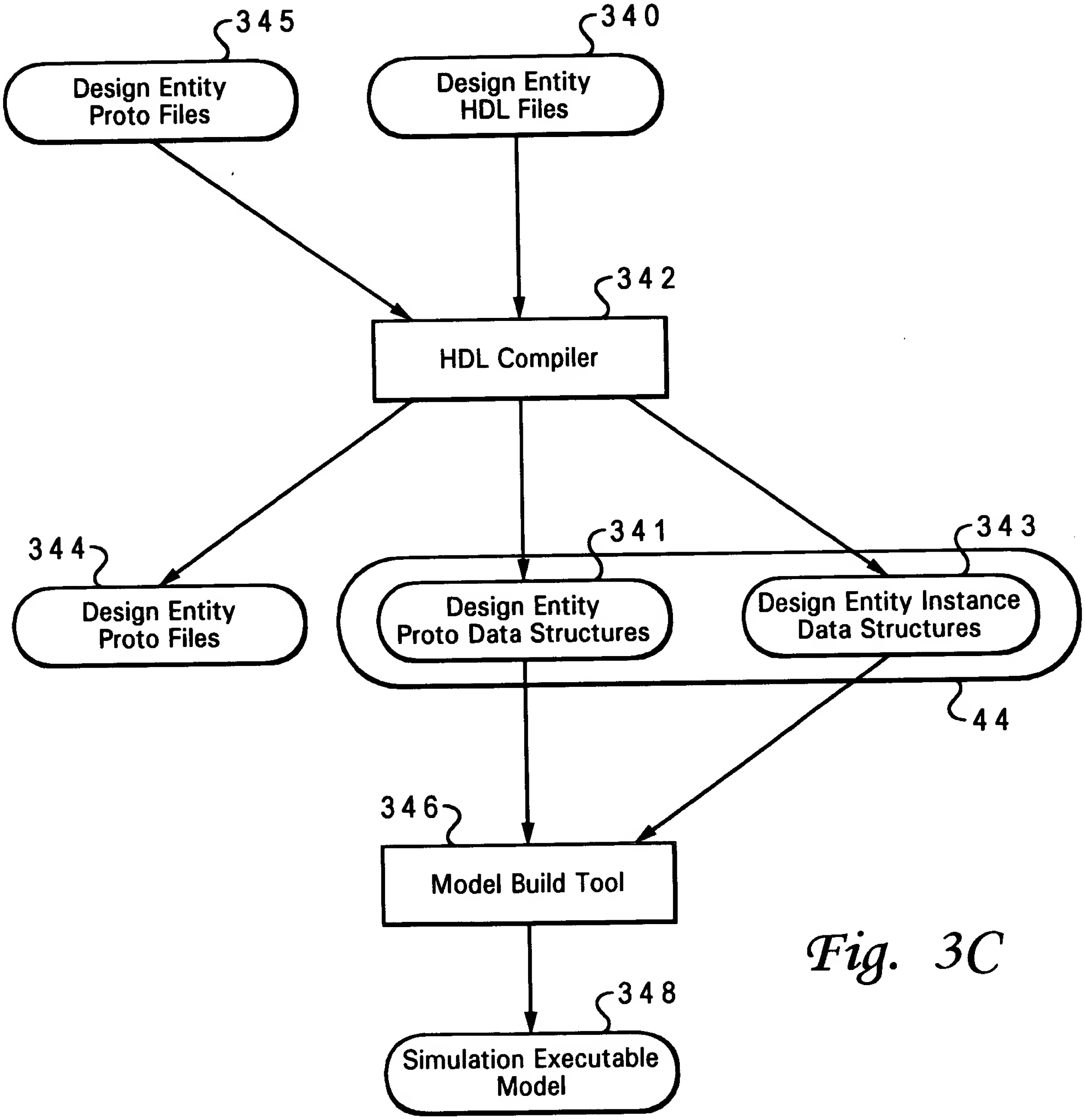

345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

Fig. 3C

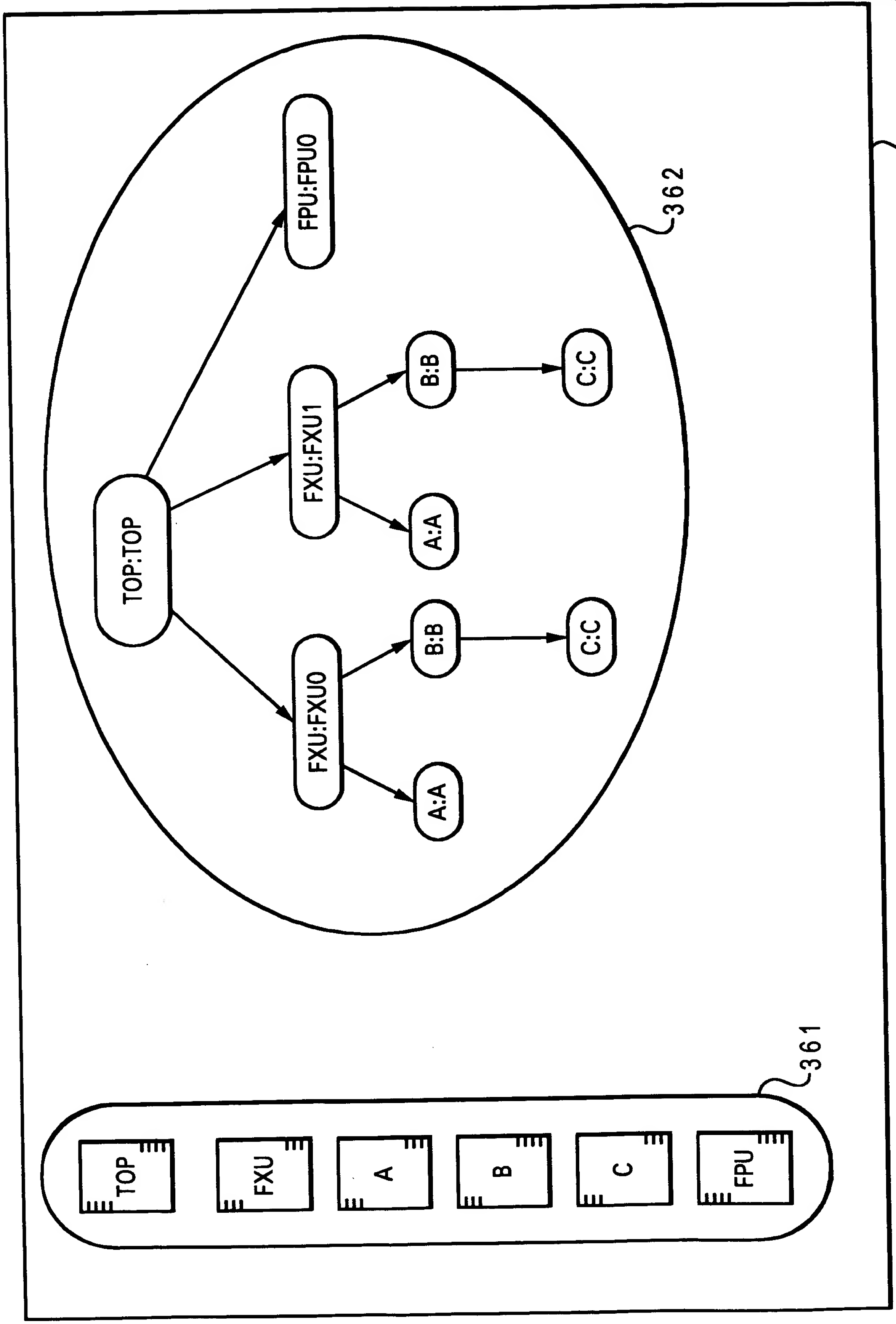

TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
B:B
C:C
A:A
B:B
C:C
362
44
TOP
FXU
A
B
C
FPU
361

Fig. 3D

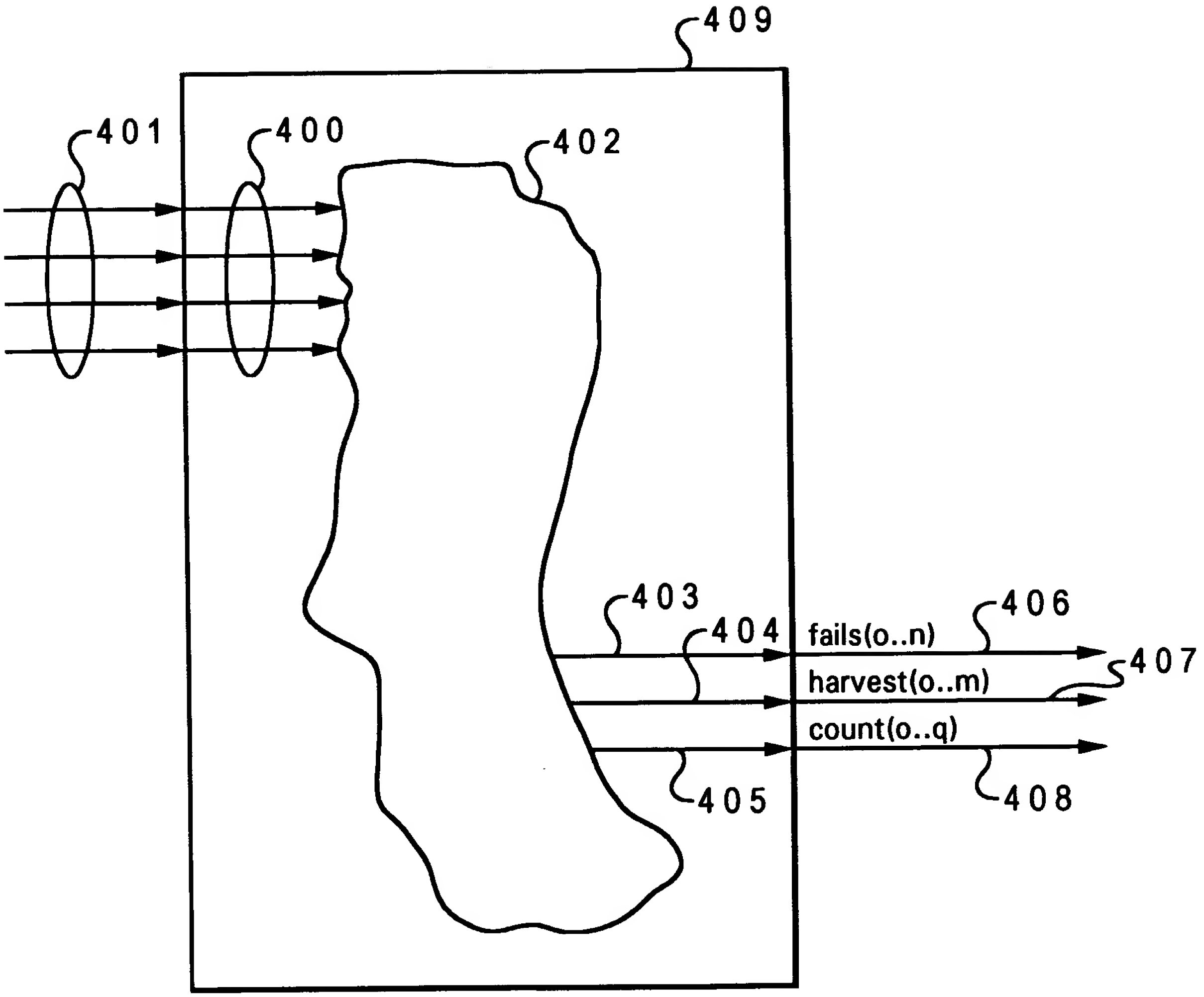

409
401
400
402
403
404
405
406
407
408
fails(o..n)
harvest(o..m)
count(o..q)

Fig. 4A

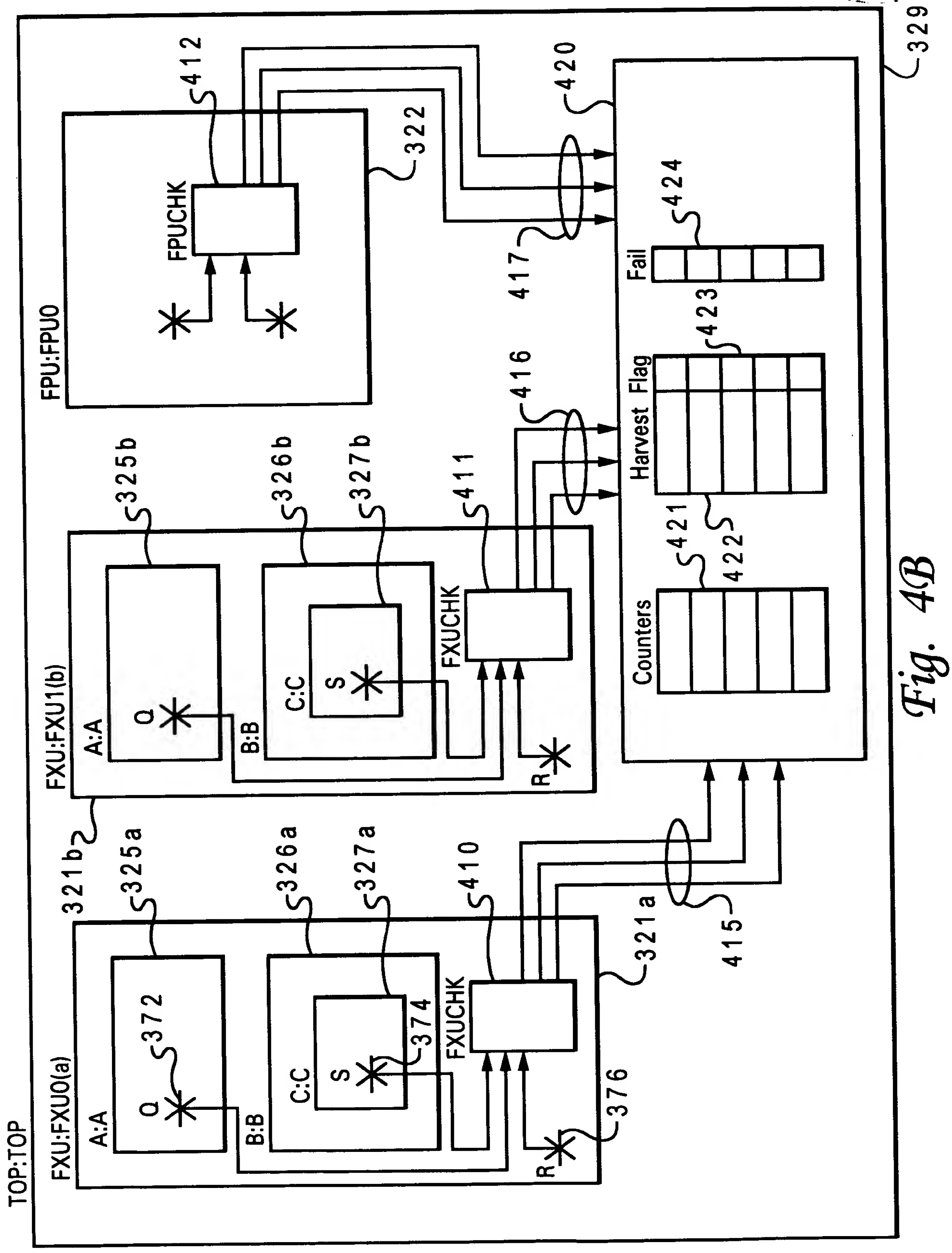
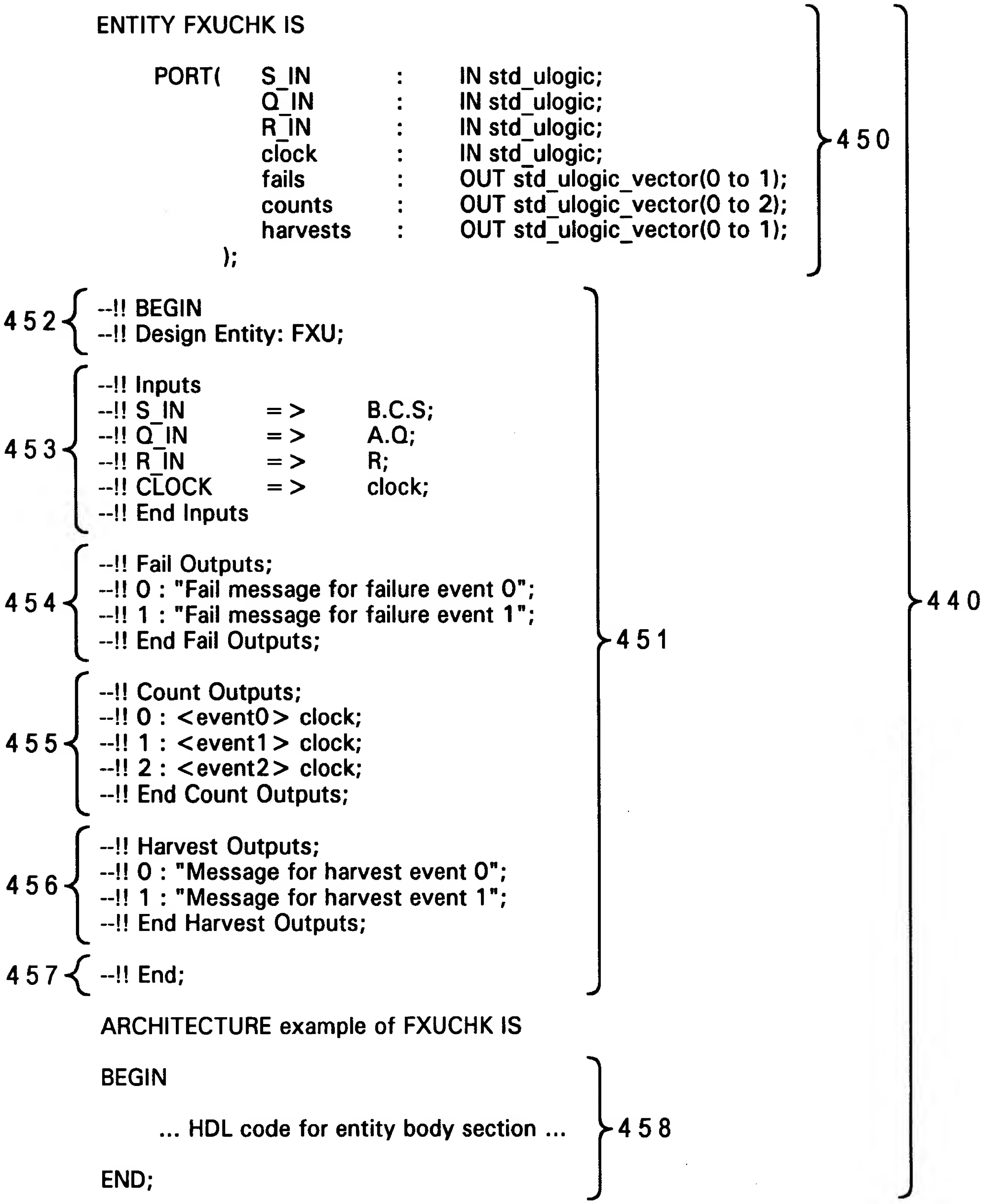

TOP:TOP
FXU:FXU0(a)
A:A
Q
372
B:B
C:C
S
374
FXUCHK
R
376
325a
326a
327a
410
321a
321b
415
FXU:FXU1(b)
A:A
Q
B:B
C:C
S
FXUCHK
R
325b
326b
327b
411
416
FPU:FPU0
FPUCHK
412
322
417
420
Counters
421
422
Harvest
Flag
423
Fail
424
329

Fig. 4B



```
ENTITY FXUCHK IS

      PORT(   S_IN         :      IN std_ulogic;
              Q_IN         :      IN std_ulogic;
              R_IN         :      IN std_ulogic;
              clock        :      IN std_ulogic;
              fails        :      OUT std_ulogic_vector(0 to 1);
              counts       :      OUT std_ulogic_vector(0 to 2);
              harvests     :      OUT std_ulogic_vector(0 to 1);
          );

--!! BEGIN
--!! Design Entity: FXU;

--!! Inputs
--!! S_IN         =>      B.C.S;
--!! Q_IN         =>      A.Q;
--!! R_IN         =>      R;
--!! CLOCK        =>      clock;
--!! End Inputs

--!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;

--!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;

--!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;

--!! End;

ARCHITECTURE example of FXUCHK IS

BEGIN

      ... HDL code for entity body section ...

END;
```

Fig. 4C

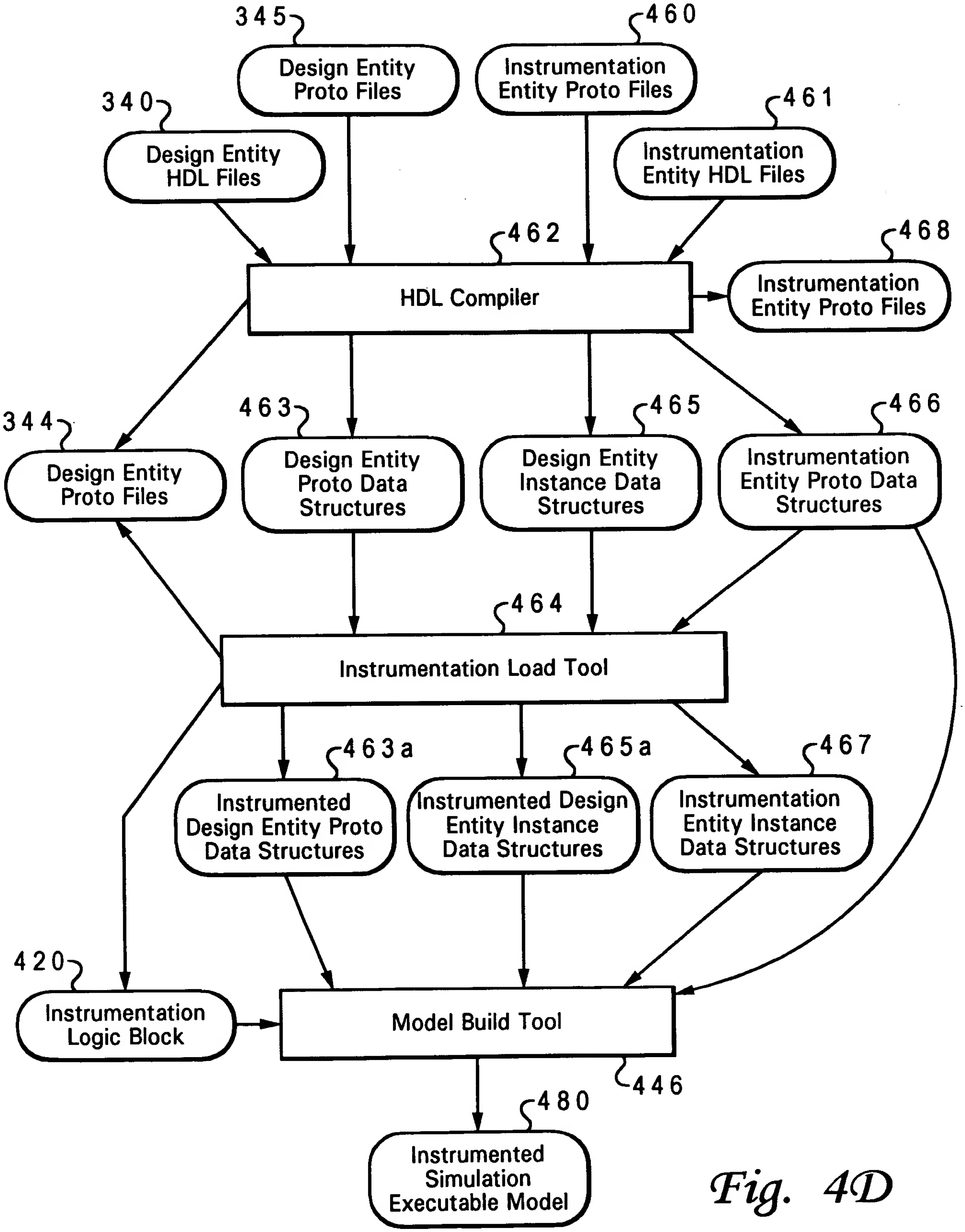

345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

Fig. 4D

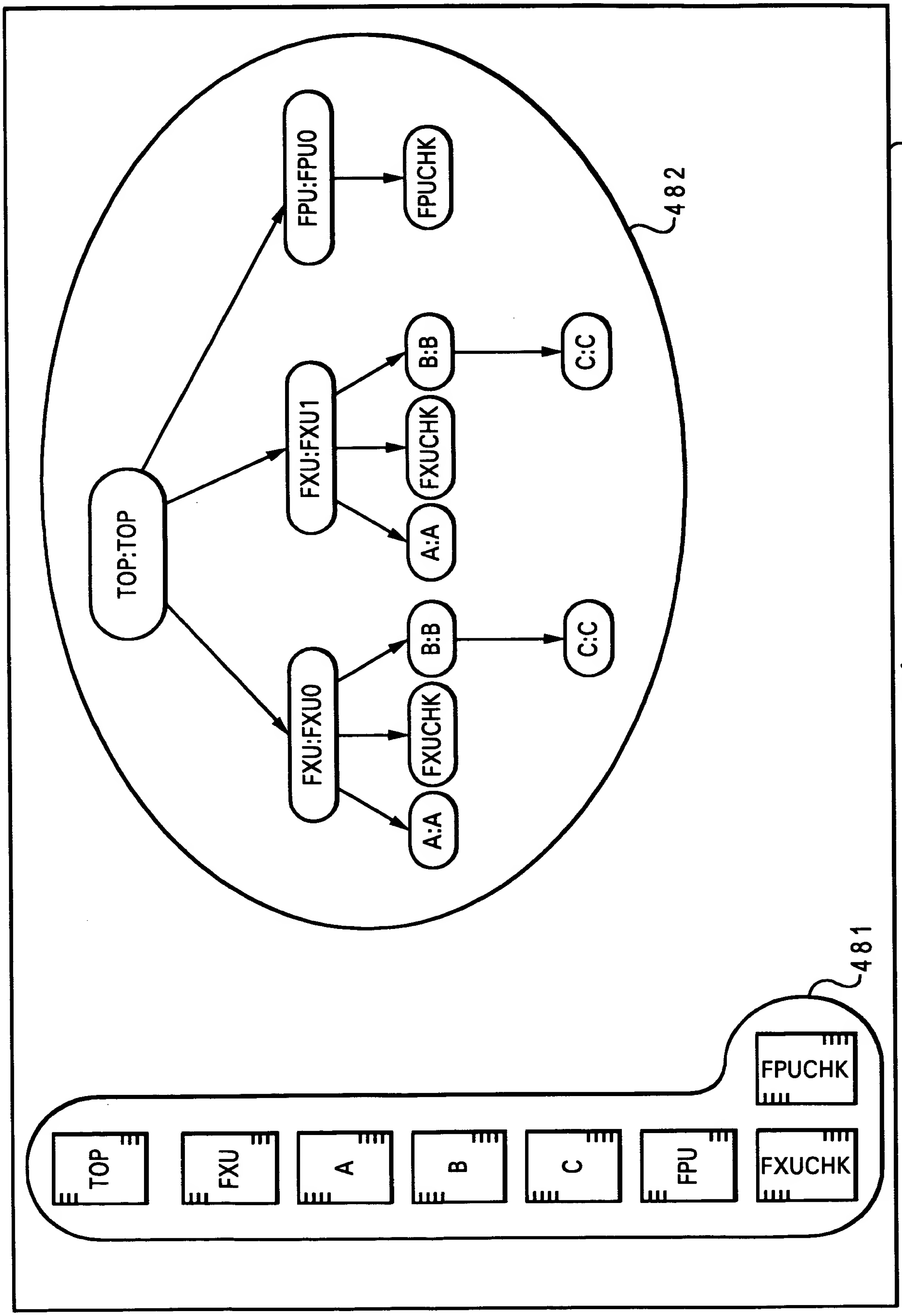
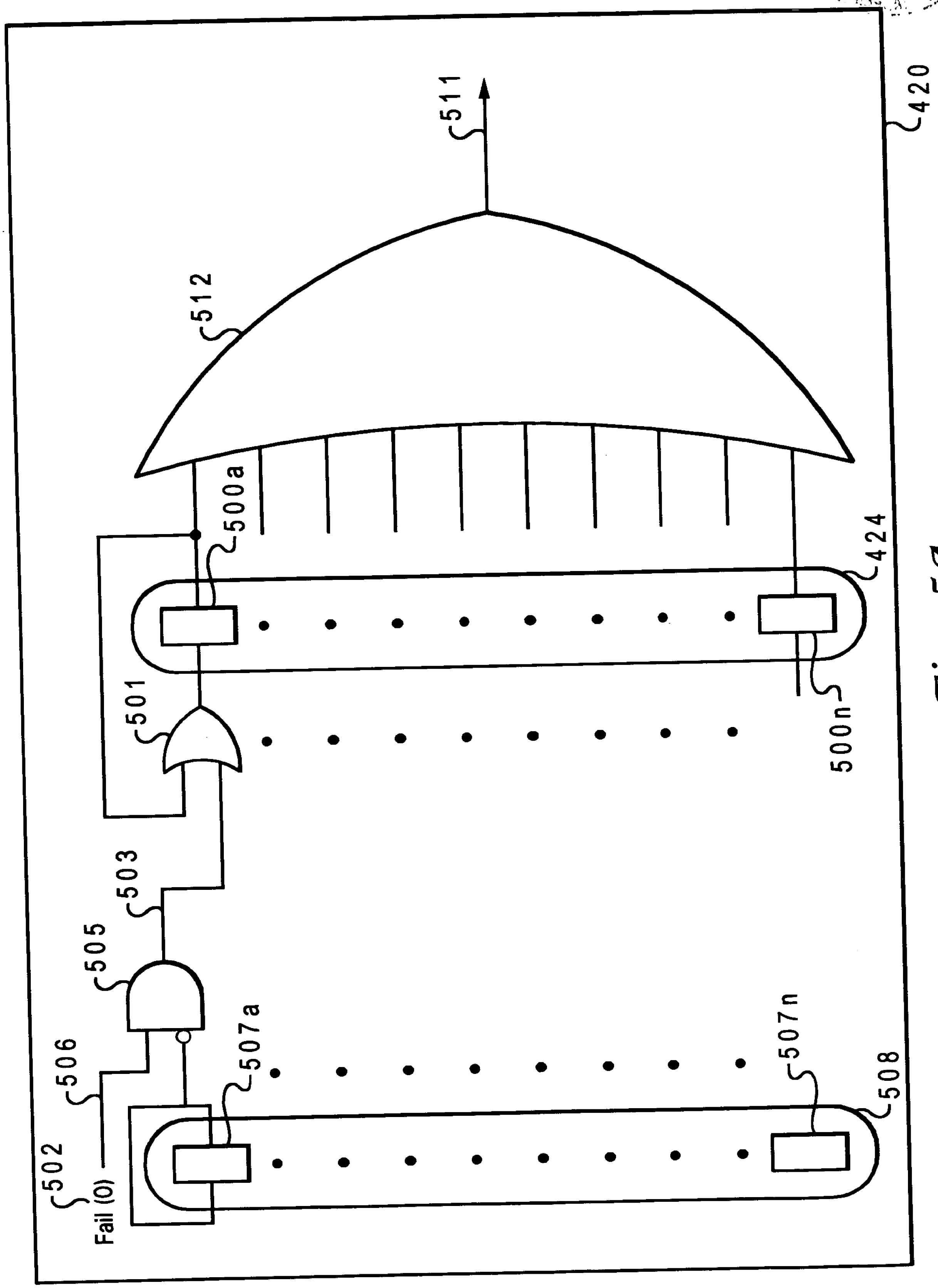

TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
A:A
FXUCHK
B:B
C:C
FPUCHK
482
44
481
TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK

Fig. 4E



511
420
512
500a
424
501
500n
503
505
506
507a
507n
508
502
Fail (0)

Fig. 5A

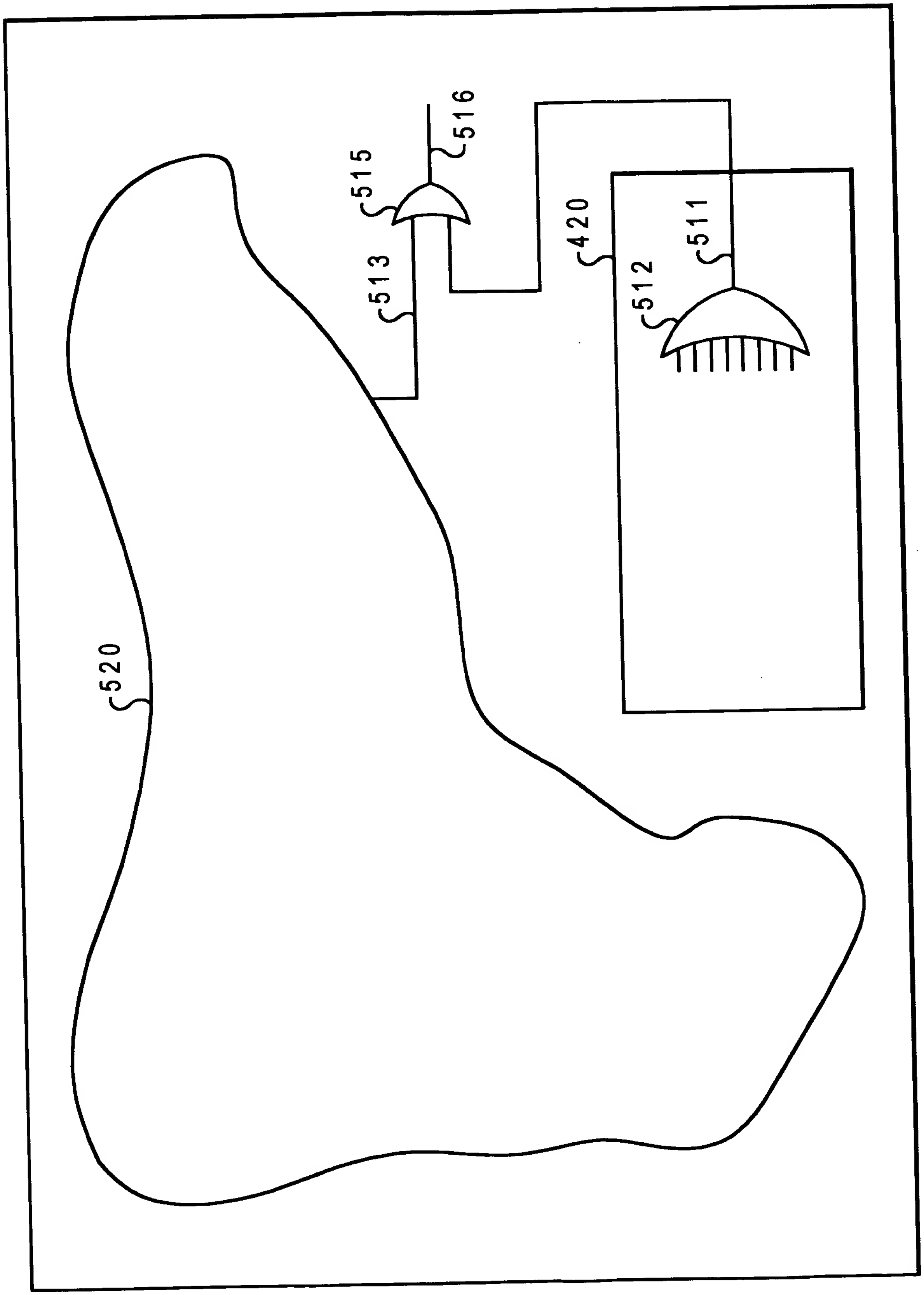


Fig. 5B

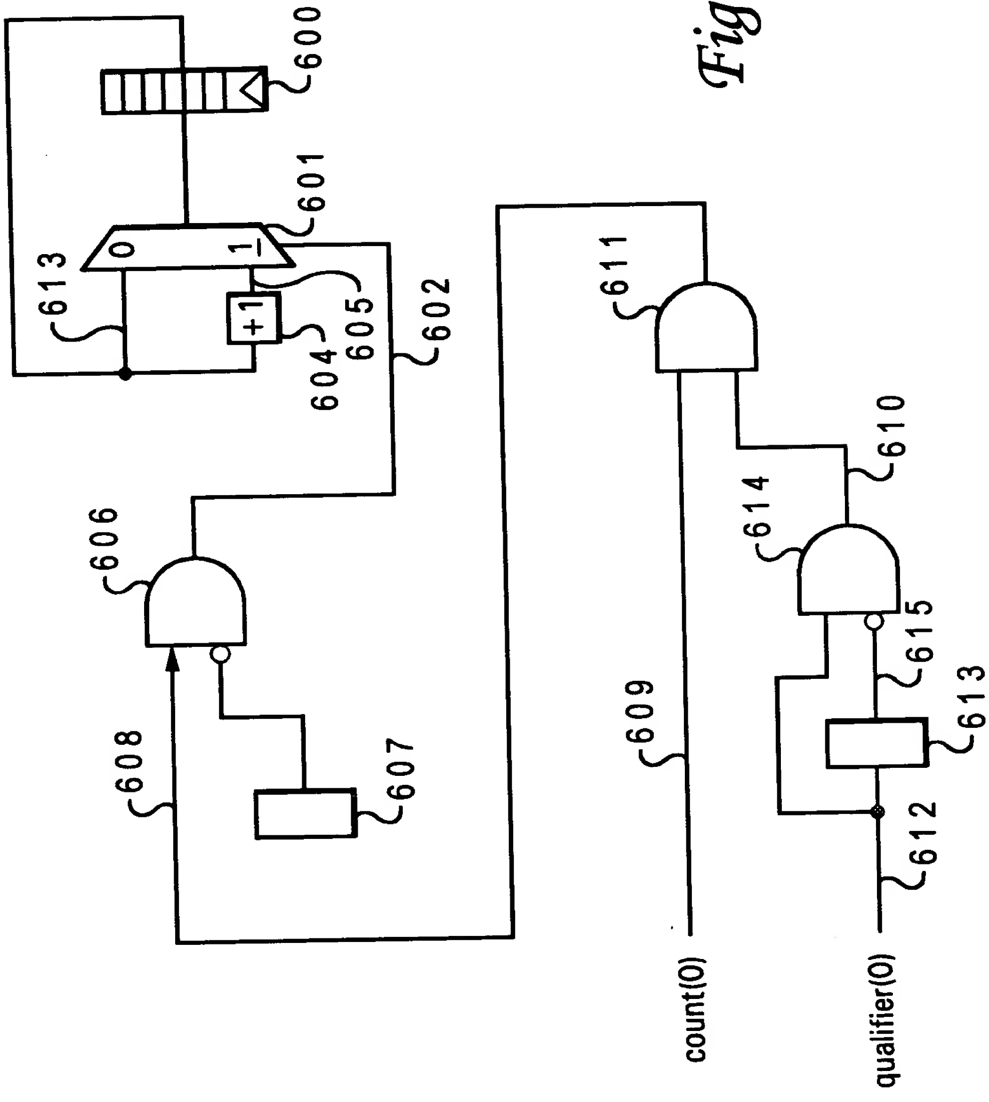

600
601
0
1
613
+1
605
604
602
606
608
607
611
610
614
615
613
609
612
count(0)
qualifier(0)

Fig. 6A

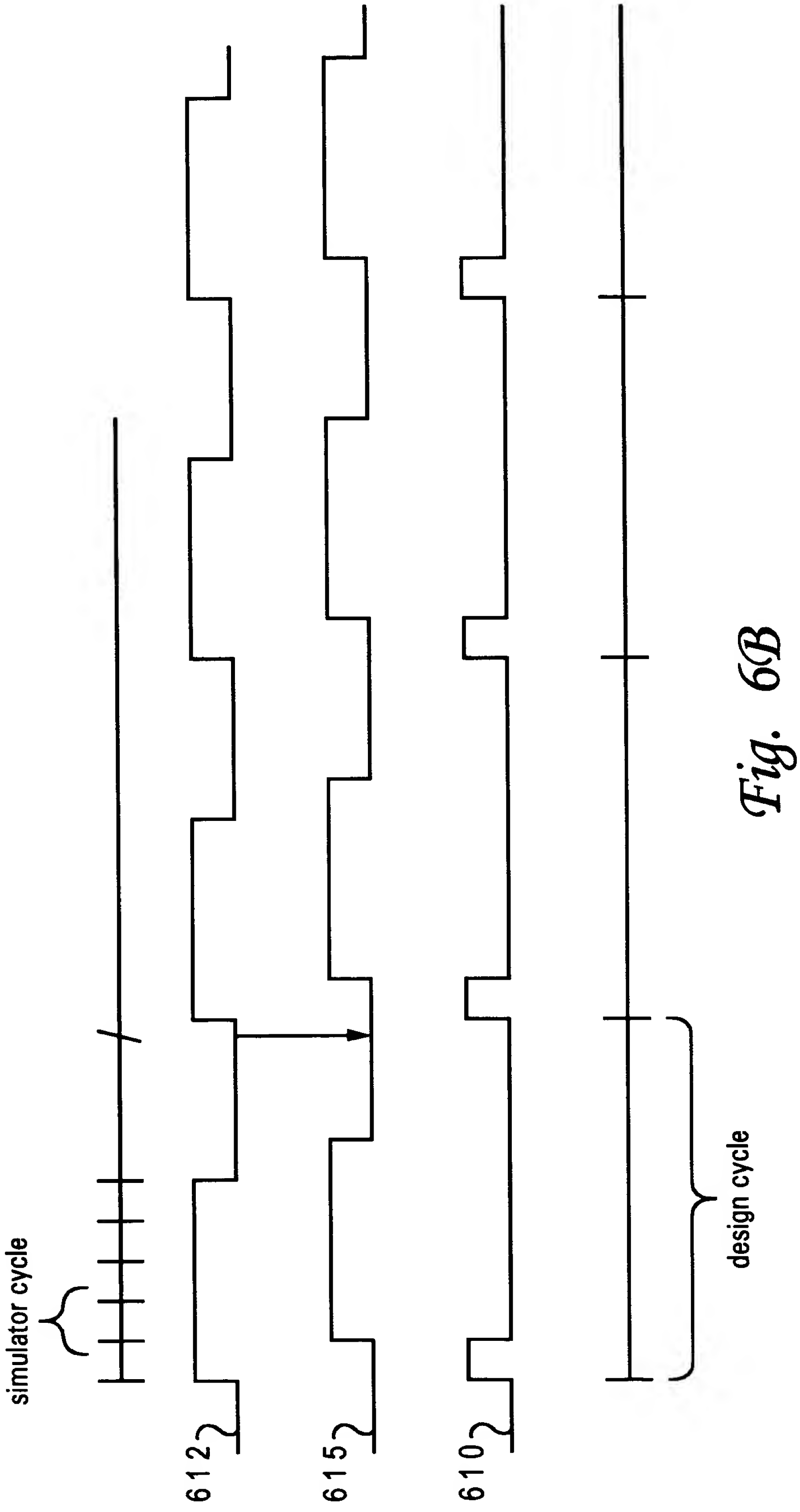


Fig. 6B

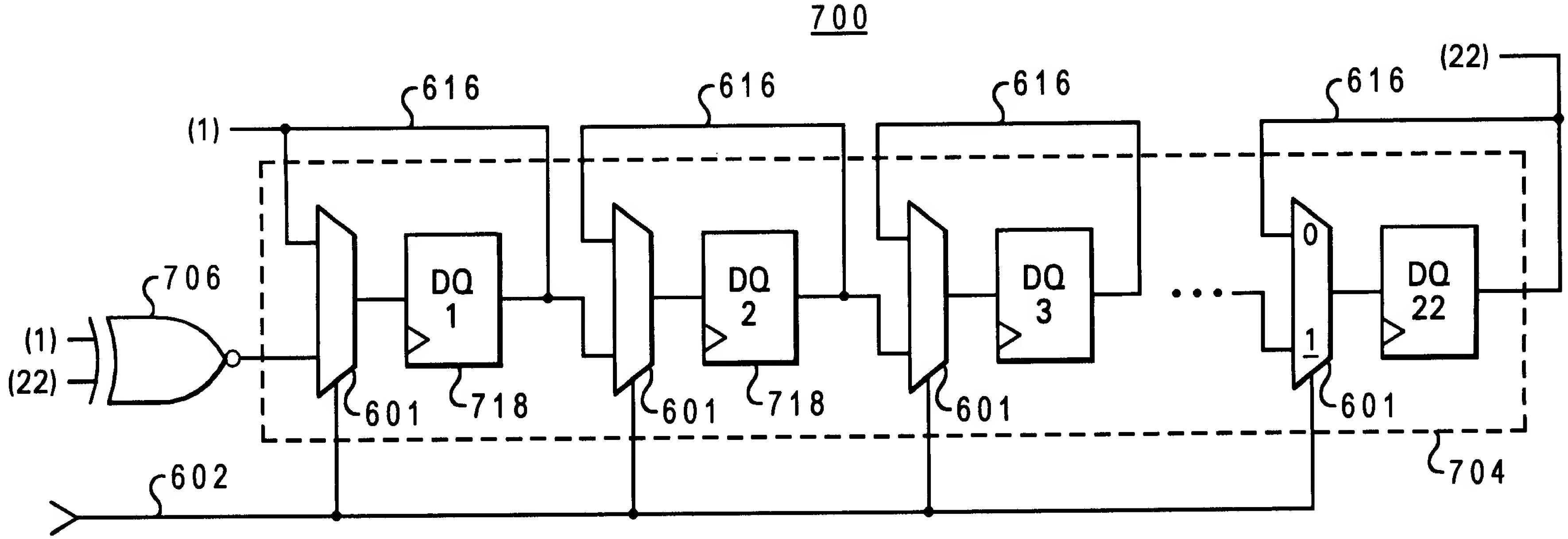

700
(1)
616
616
616
616
(22)
706
(1)
(22)
DQ
1
DQ
2
DQ
3
DQ
22
0
1
601
718
601
718
601
601
704
602

Fig. 7

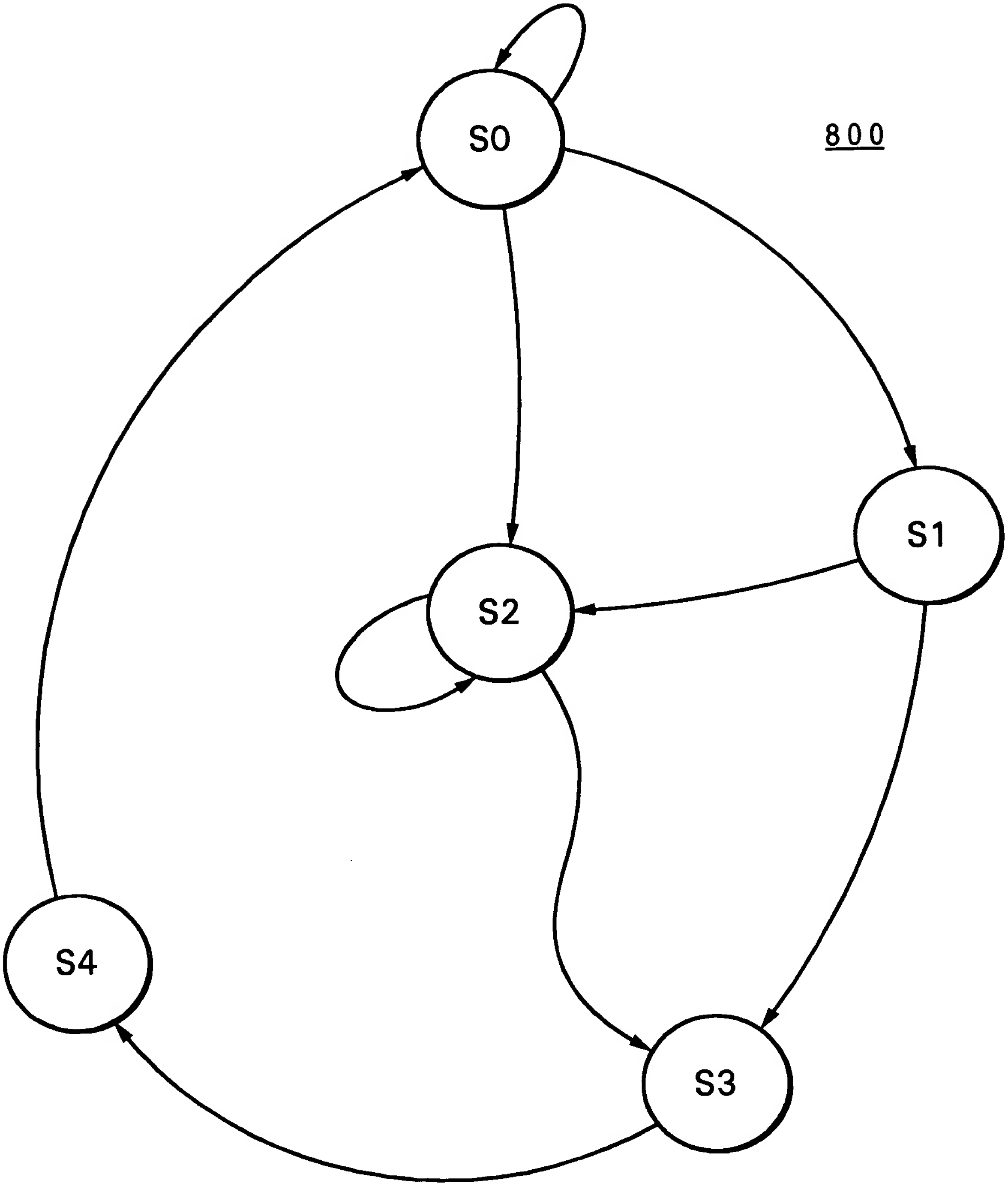


Fig. 8A
Prior Art

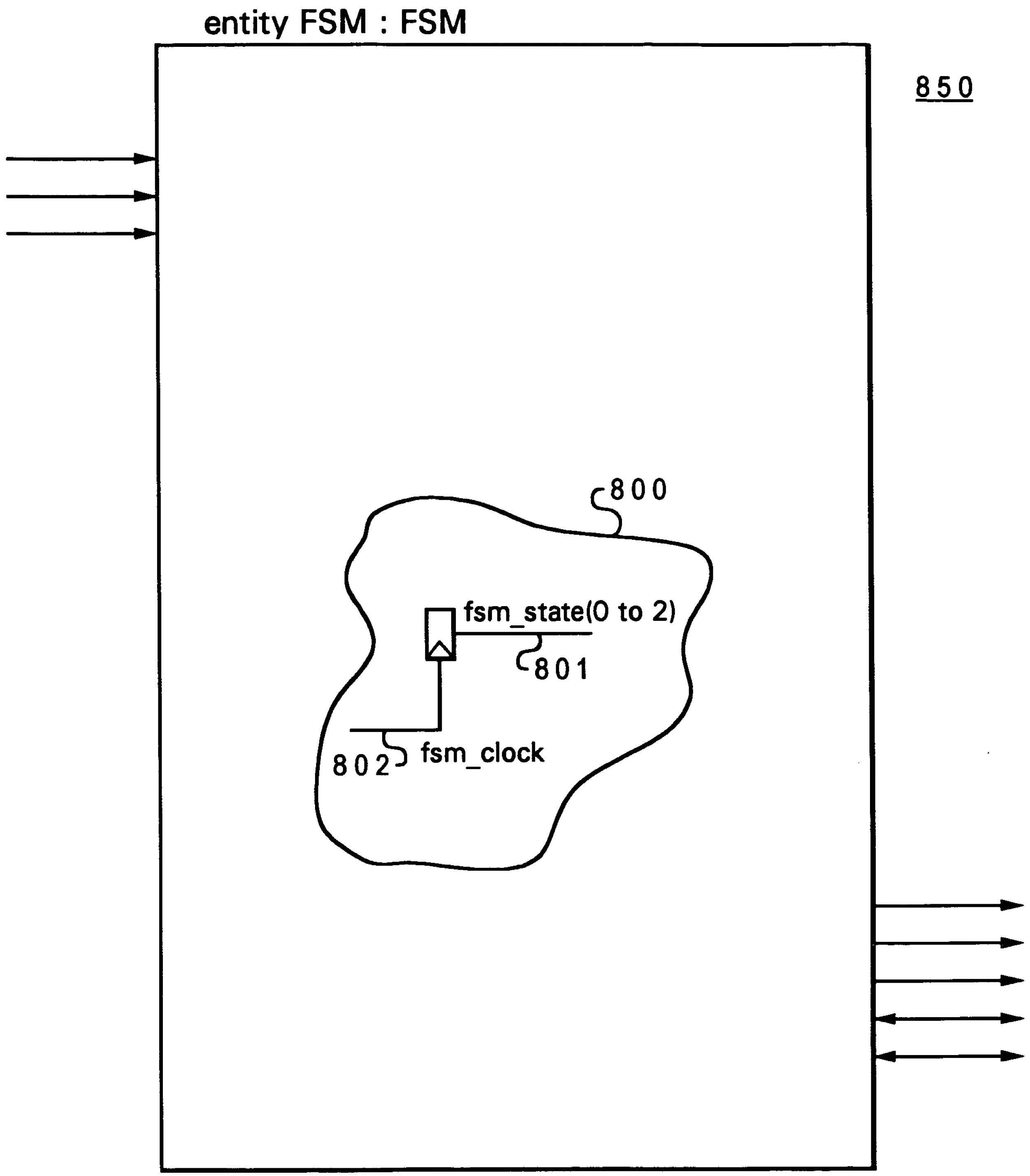


Fig. 8B
Prior Art

```
ENTITY FSM IS

      PORT(
                  ....ports for entity fsm....
            );

      ARCHITECTURE FSM OF FSM IS

      BEGIN

                  ... HDL code for FSM and rest of the entity ...

                  fsm_state(0 to 2) <= ... Signal 801 ...

      853{ --!! Embedded FSM  : examplefsm;                              }         }
      859{ --!! clock         : (fsm_clock);                             }         }
      854{ --!! state_vector  : (fsm_state(0 to 2));                     }         }
      855{ --!! states        : (S0, S1, S2, S3, S4);                    }852      }860
      856{ --!! state_encoding : ('000', '001', '010', '011', '100');     }         }
         { --!! arcs          : (S0 => S0, S0 => S1, S0 => S2,           }         }
      857{ --!!                  (S1 => S2, S1 => S3, S2 => S2,           }         }
         { --!!                  (S2 => S3, S3 => S4, S4 => S0);          }         }
      858{ --!! End FSM;                                                           }
                        ,
                        ,
                        ,
                        ,
                        ,
                        ,
                        ,

      END;
```

Fig. 8C

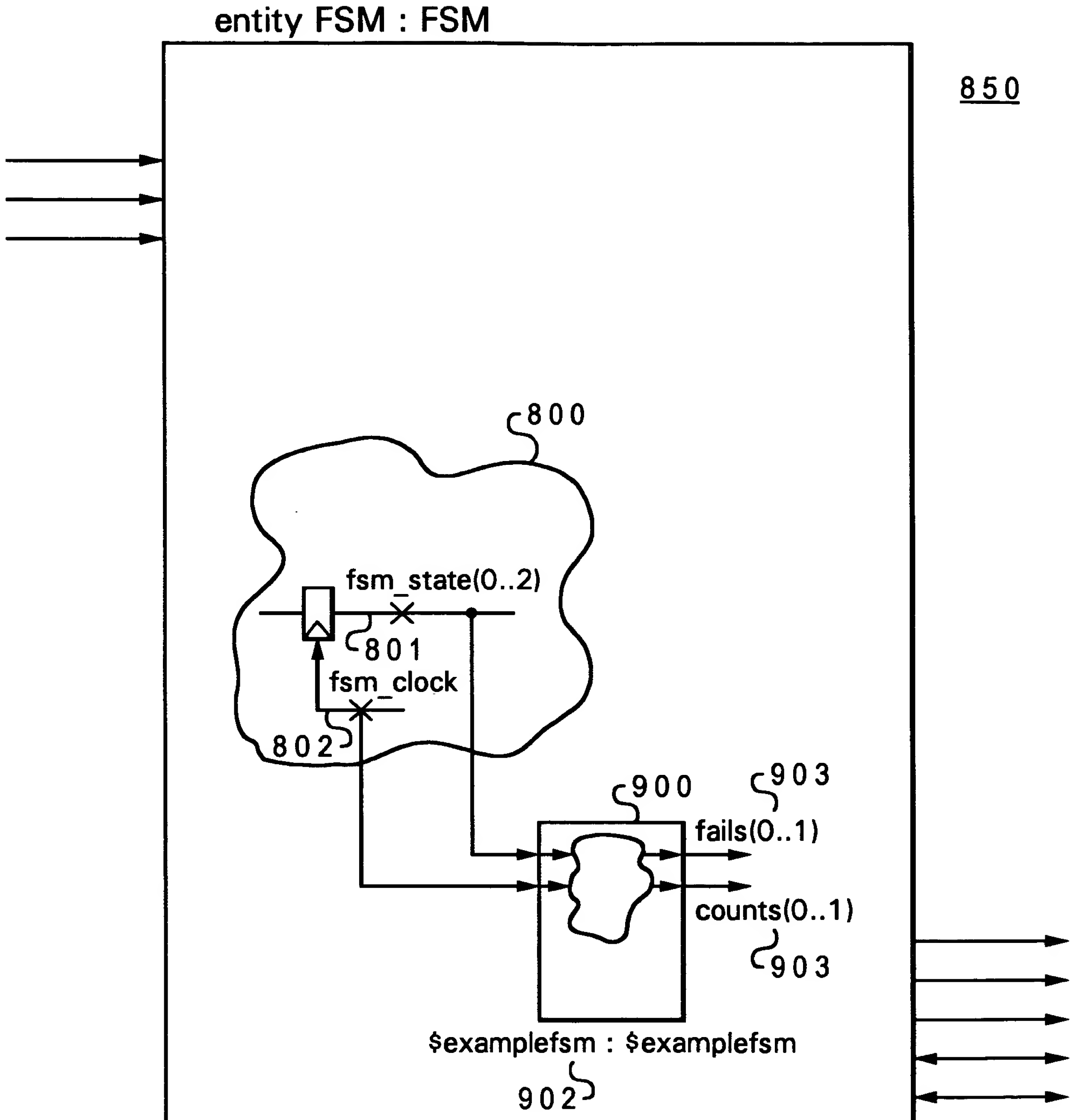

entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

Fig. 9

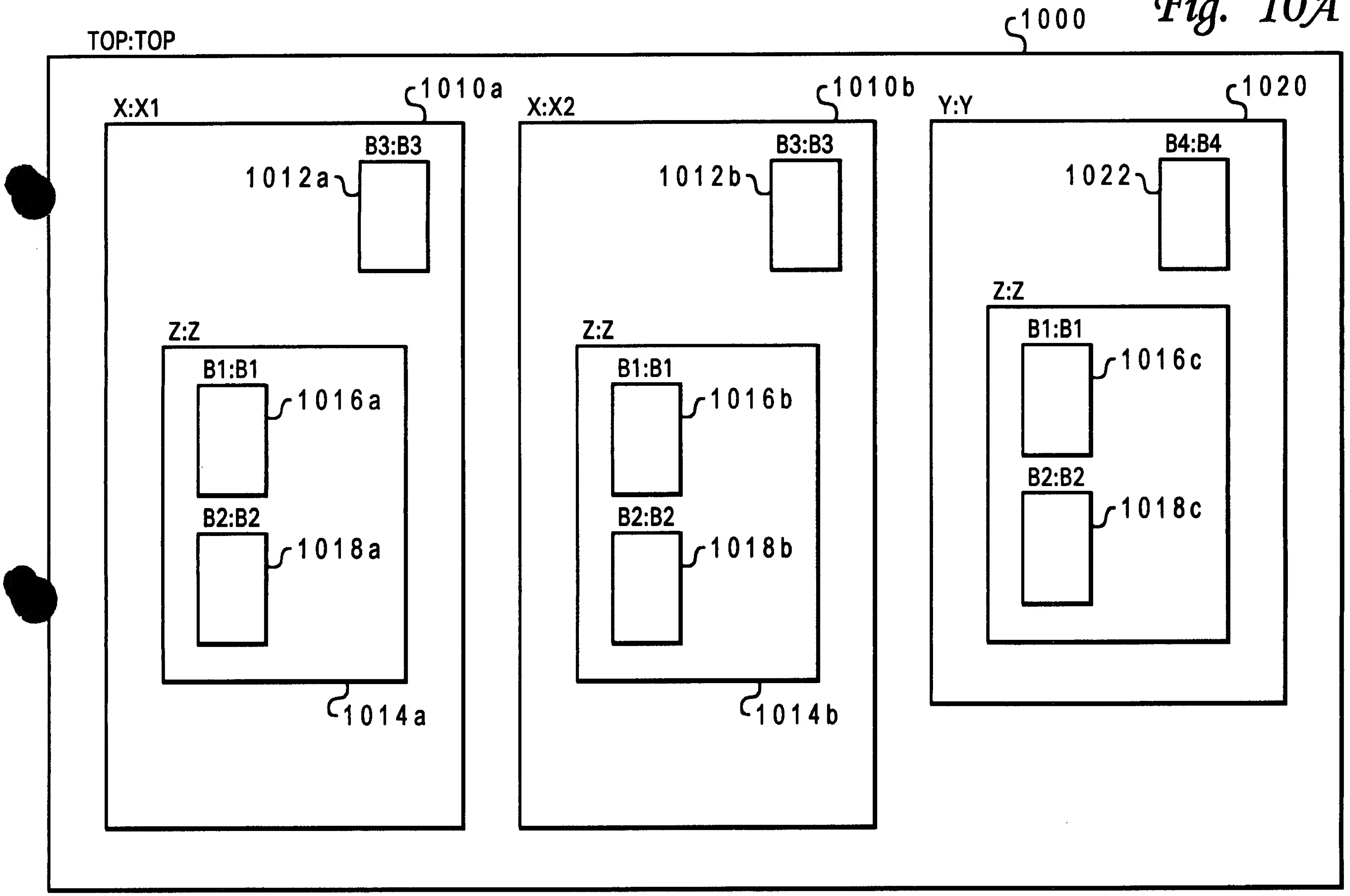

1000
TOP:TOP
X:X1
1010a
B3:B3
1012a
Z:Z
B1:B1
1016a
B2:B2
1018a
1014a
X:X2
1010b
B3:B3
1012b
Z:Z
B1:B1
1016b
B2:B2
1018b
1014b
Y:Y
1020
B4:B4
1022
Z:Z
B1:B1
1016c
B2:B2
1018c

Fig. 10A

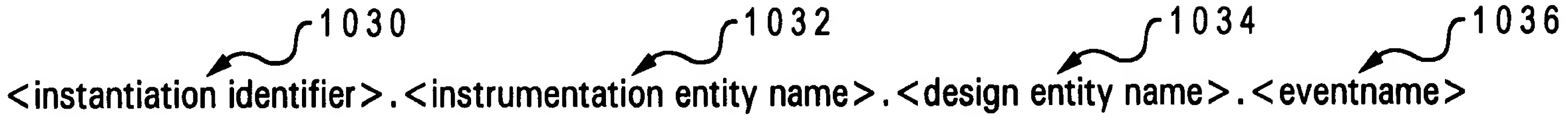
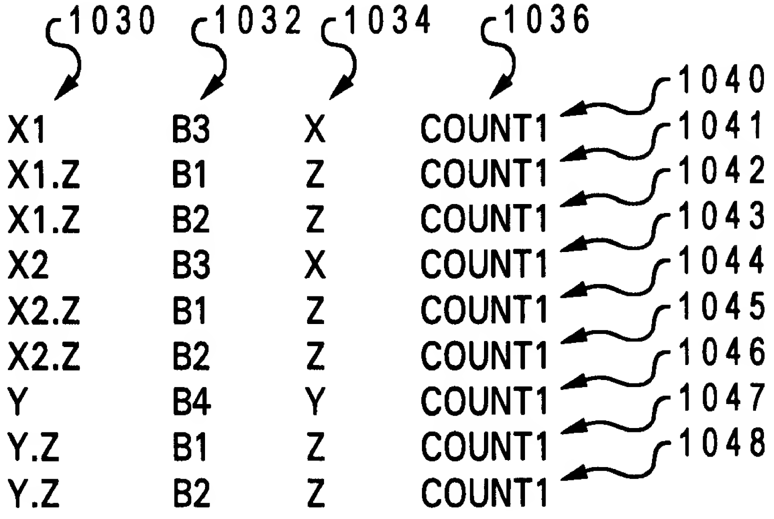

1030
1032
1034
1036
<instantiation identifier>.<instrumentation entity name>.<design entity name>.<eventname>

Fig. 10B


1030
1032
1034
1036
1040
1041
1042
1043
1044
1045
1046
1047
1048

X1	B3	X	COUNT1
X1.Z	B1	Z	COUNT1
X1.Z	B2	Z	COUNT1
X2	B3	X	COUNT1
X2.Z	B1	Z	COUNT1
X2.Z	B2	Z	COUNT1
Y	B4	Y	COUNT1
Y.Z	B1	Z	COUNT1
Y.Z	B2	Z	COUNT1

Fig. 10C

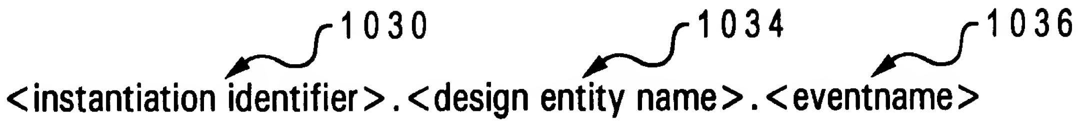


Fig. 10D

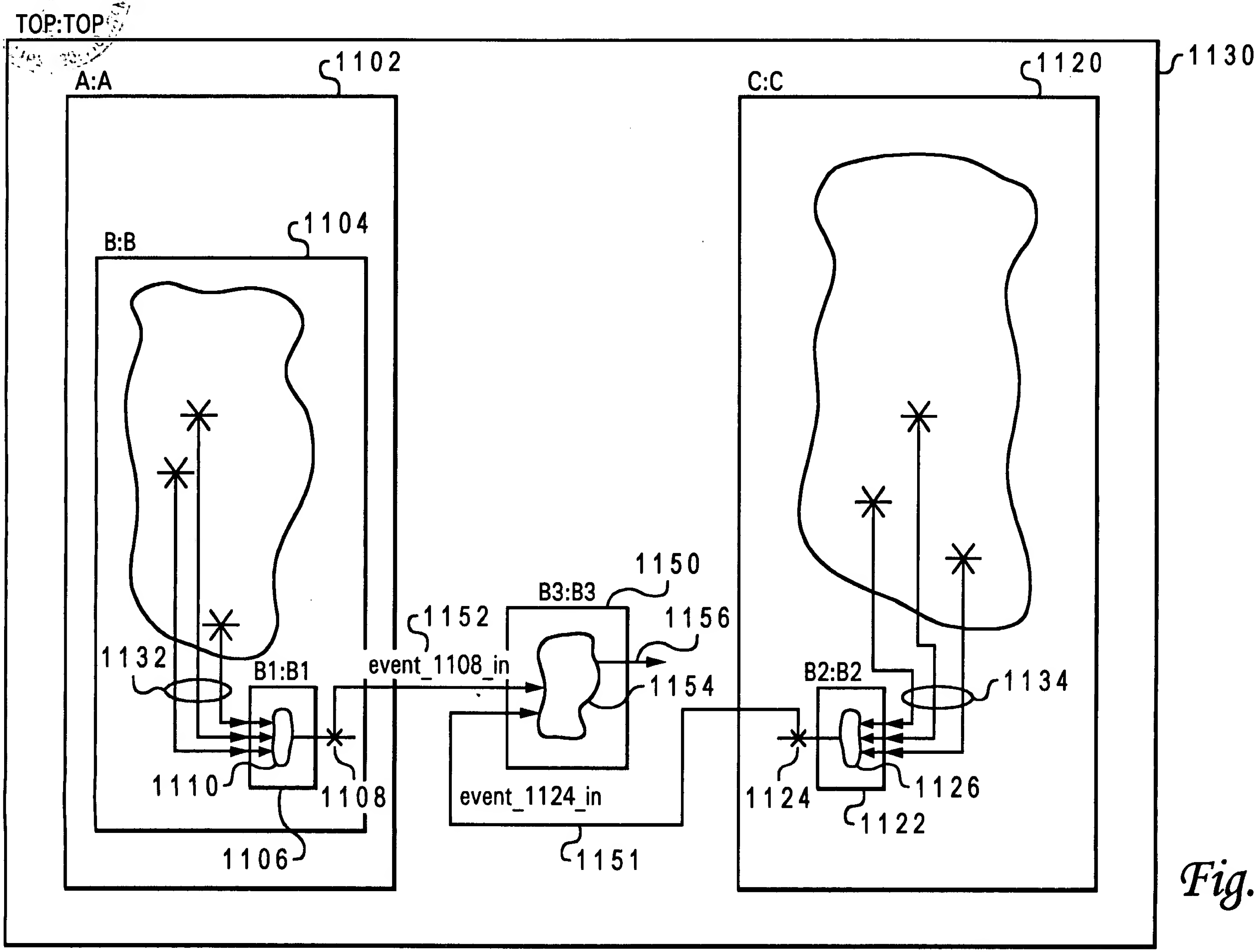
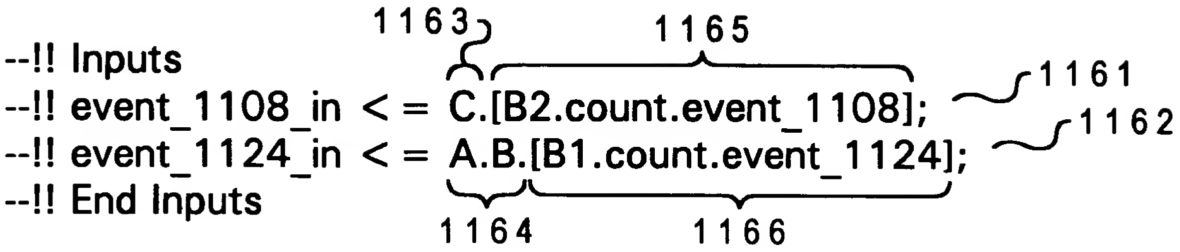
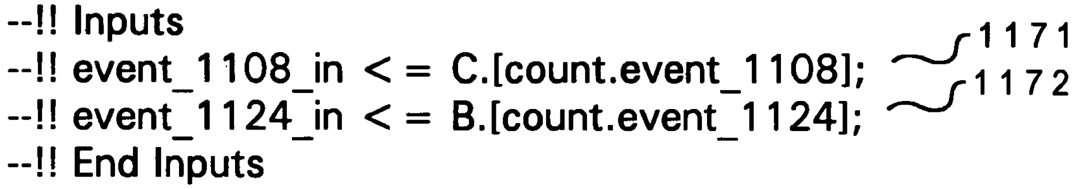

TOP:TOP
1130
A:A
1102
B:B
1104
C:C
1120
1132
B1:B1
1110
1108
1106
event_1108_in
1152
B3:B3
1150
1156
1154
event_1124_in
1151
B2:B2
1134
1124
1126
1122

Fig. 11A



```
--!! Inputs
--!! event_1108_in < = C.[B2.count.event_1108];
--!! event_1124_in < = A.B.[B1.count.event_1124];
--!! End Inputs
```

Fig. 11B



```
--!! Inputs
--!! event_1108_in < = C.[count.event_1108];
--!! event_1124_in < = B.[count.event_1124];
--!! End Inputs
```

Fig. 11C

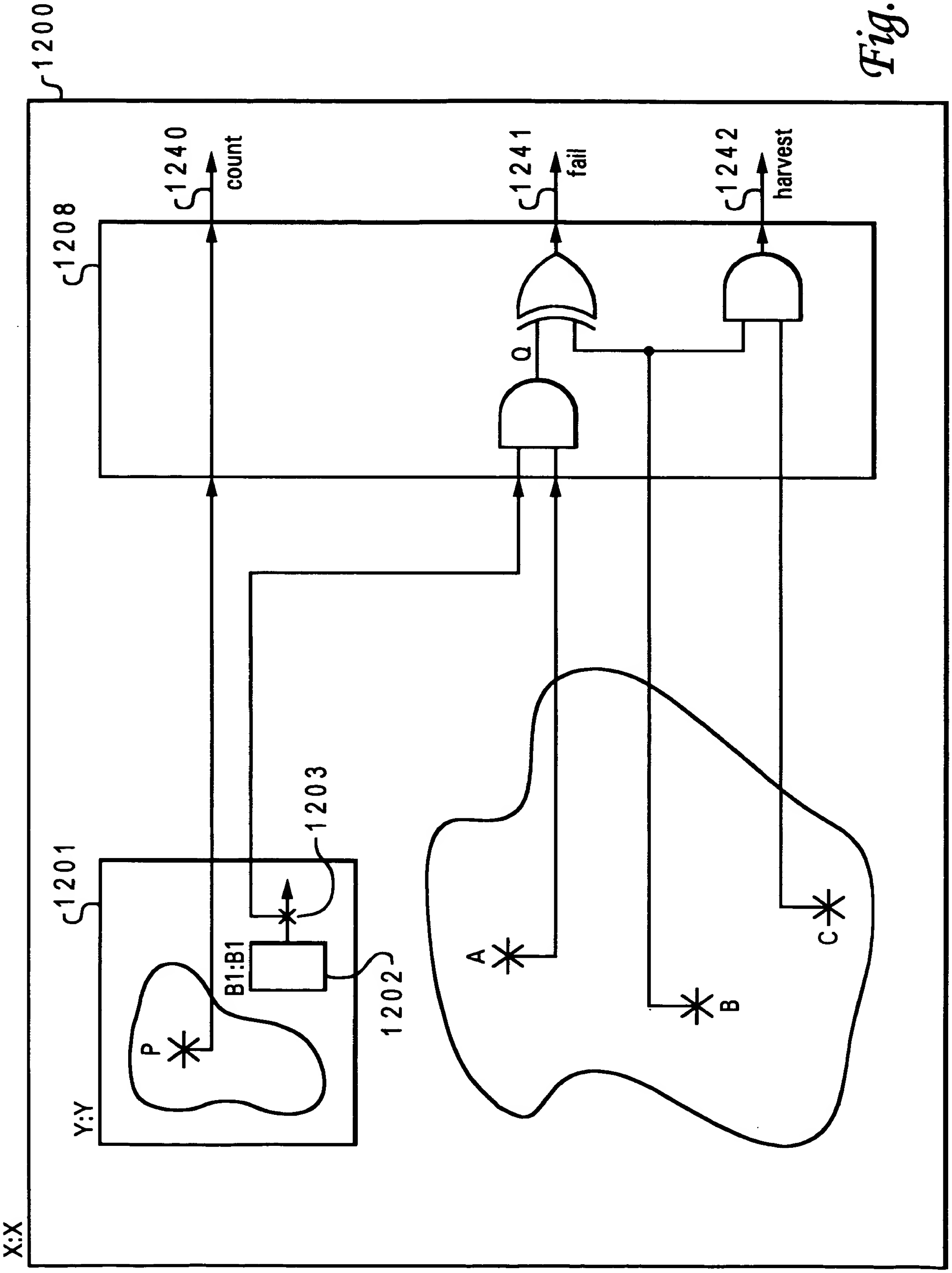

1200
X:X
1201
Y:Y
P
B1:B1
1202
1203
1208
Q
1240
count
1241
fail
1242
harvest
A
B
C

Fig. 12A

```
          ENTITY X IS
                PORT(    :
                         :
                         :
                     );
          ARCHITECTURE example of X IS
          BEGIN
                .
                .
                .
                .
                .
             ... HDL code for X ...
                .
                .
                .
                .
                .                                                          1220
       {  Y:Y
  1221 {  PORT MAP(   :
       {              :
       {          );
       {  A <= ....
  1222 {  B <= ....
       {  C <= ....
                                              1230
       {  --!! [count, countname0, clock] <= Y.P;     1232
  1223 {  --!! Q <= Y. [B1.count.count1] AND A;          1234
       {  --!! [fail, failname0, "fail msg"] <= Q XOR B;
       {  --!! [harvest, harvestname0, "harvest msg"] <= B AND C;
          END;                                1236
```

Fig. 12B

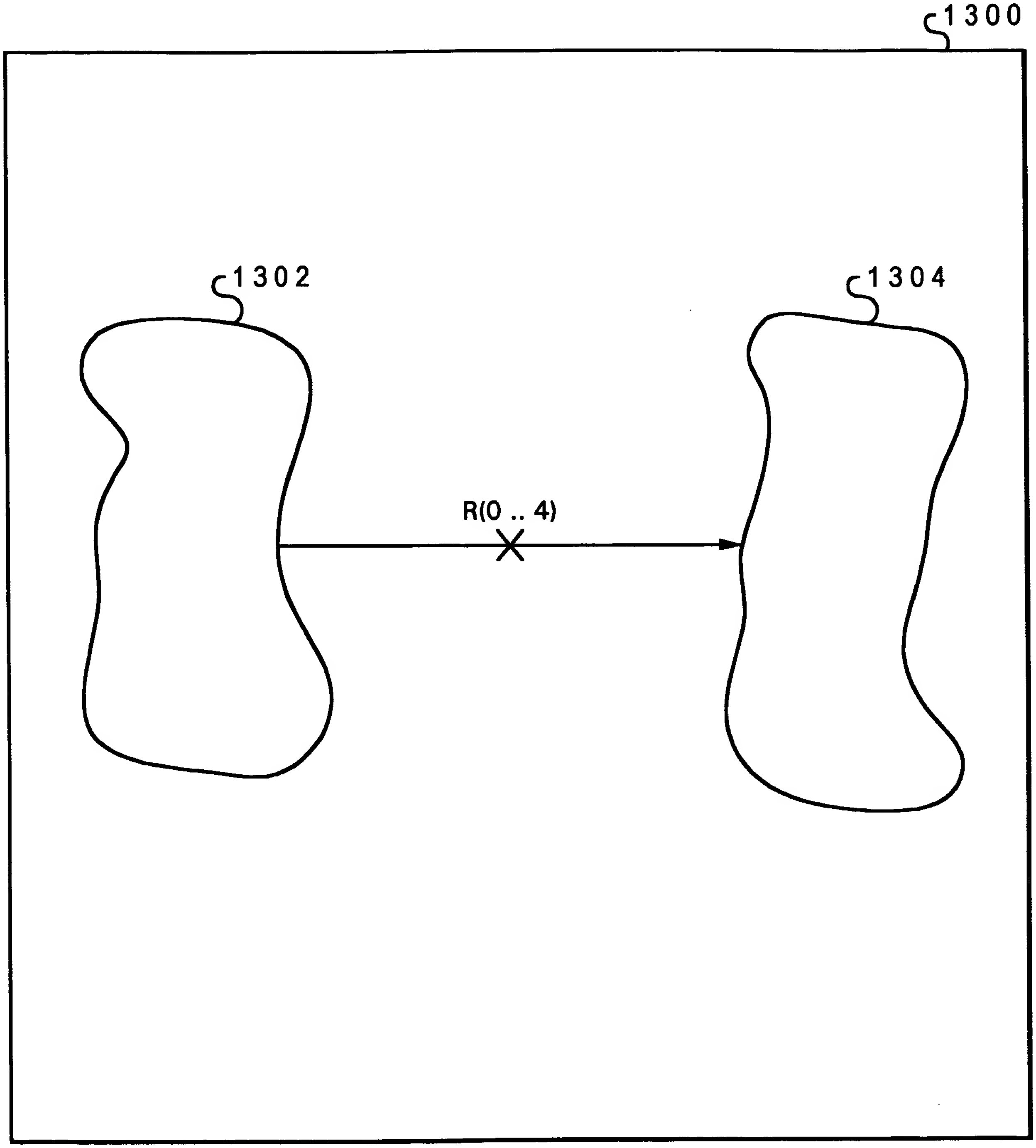


Fig. 13A

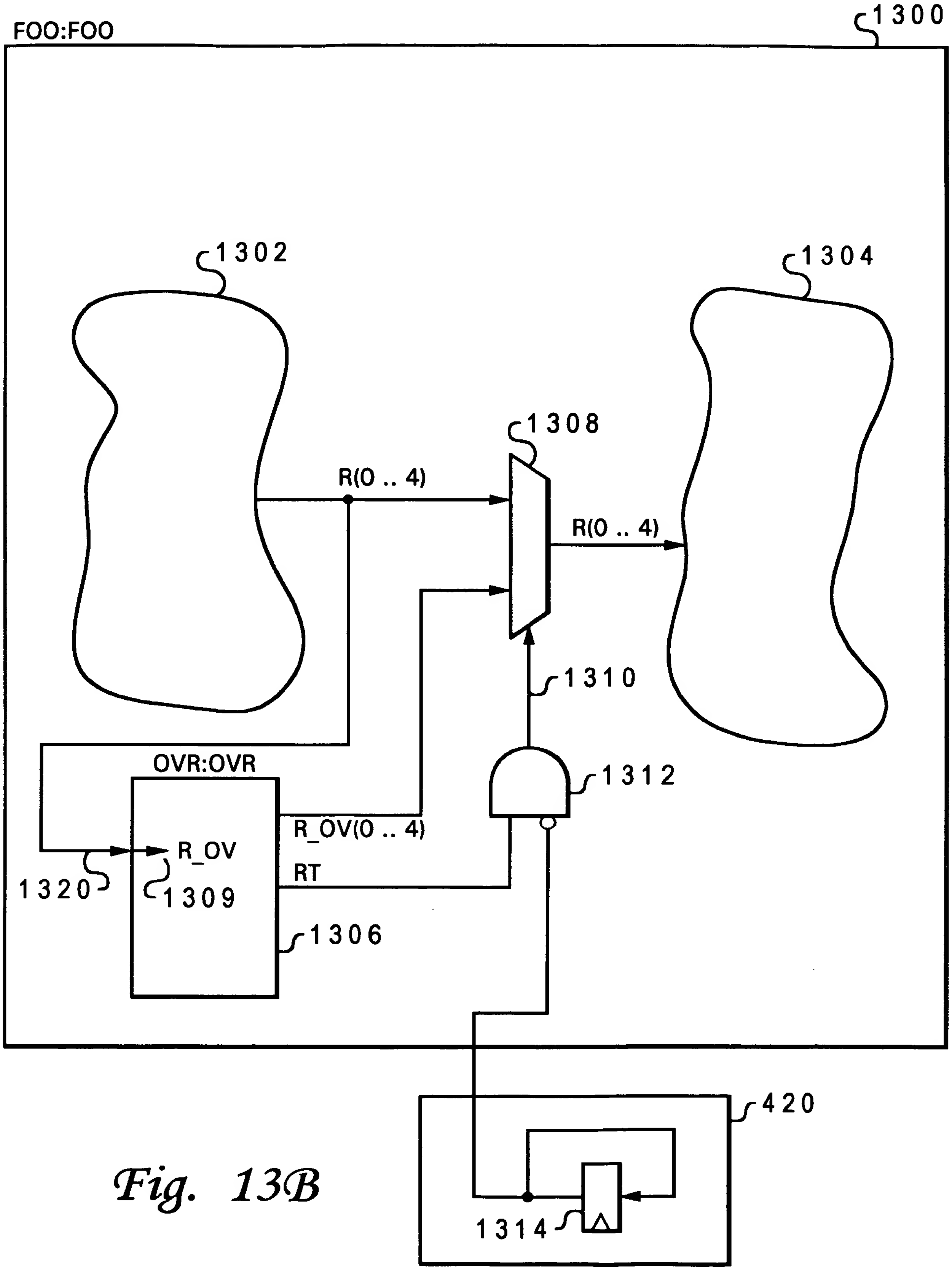
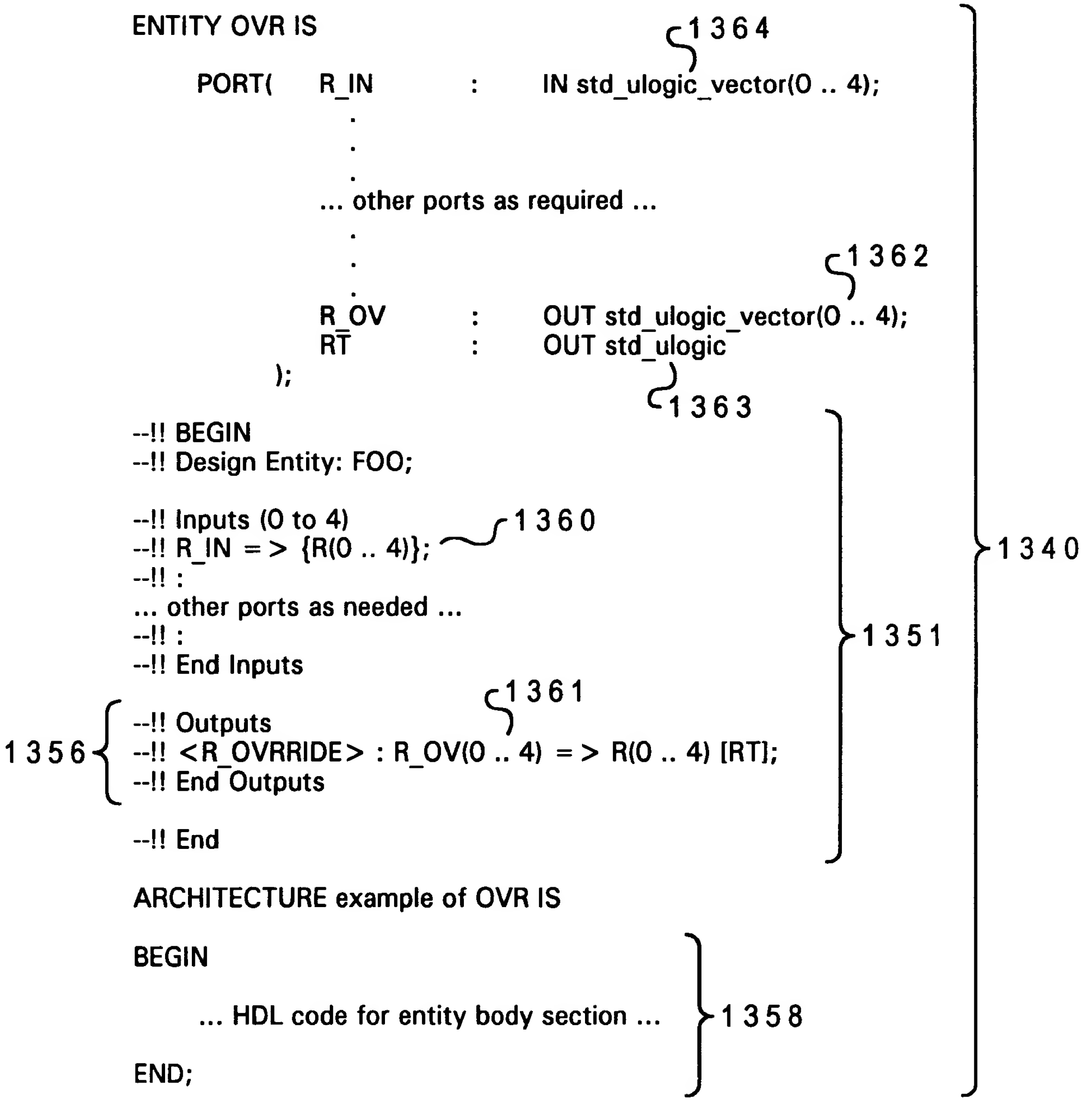

FOO:FOO
1300
1302
1304
1308
R(0 .. 4)
R(0 .. 4)
1310
OVR:OVR
1312
R_OV(0 .. 4)
R_OV
RT
1320
1309
1306
420
1314

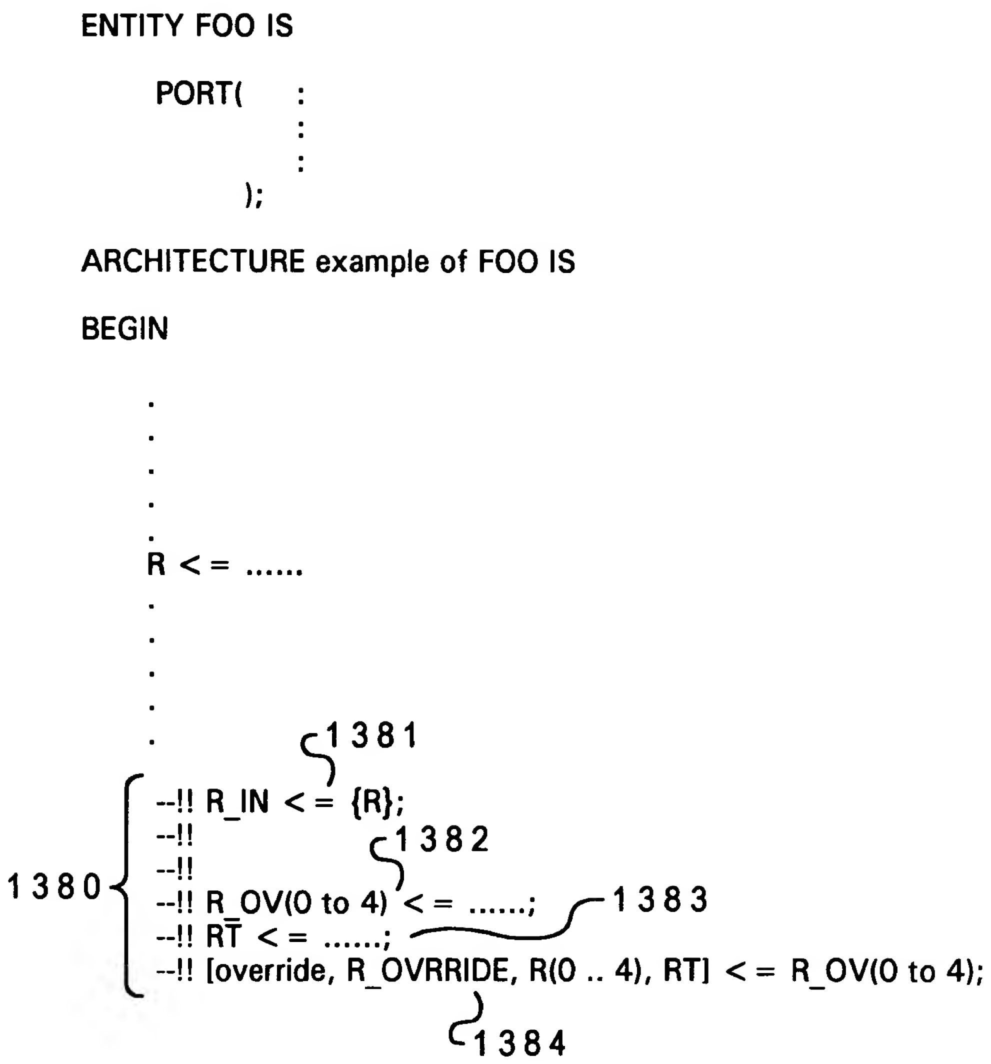
Fig. 13B



```
ENTITY OVR IS

      PORT(    R_IN          :      IN std_ulogic_vector(0 .. 4);
                  .
                  .
                  .
               ... other ports as required ...
                  .
                  .
                  .
               R_OV          :      OUT std_ulogic_vector(0 .. 4);
               RT            :      OUT std_ulogic
            );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN => {R(0 .. 4)};
--!! :
... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVRRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS

BEGIN

      ... HDL code for entity body section ...

END;
```

1364
1362
1363
1360
1361
1356
1351
1340
1358

Fig. 13C



```
ENTITY FOO IS

      PORT(     :
                :
                :
            );

ARCHITECTURE example of FOO IS

BEGIN

      .
      .
      .
      .
      .
      R <= ......
      .
      .
      .
      .
      .
      --!! R_IN <= {R};
      --!!
      --!!
      --!! R_OV(0 to 4) <= ......;
      --!! RT <= ......;
      --!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
```

Fig. 13D

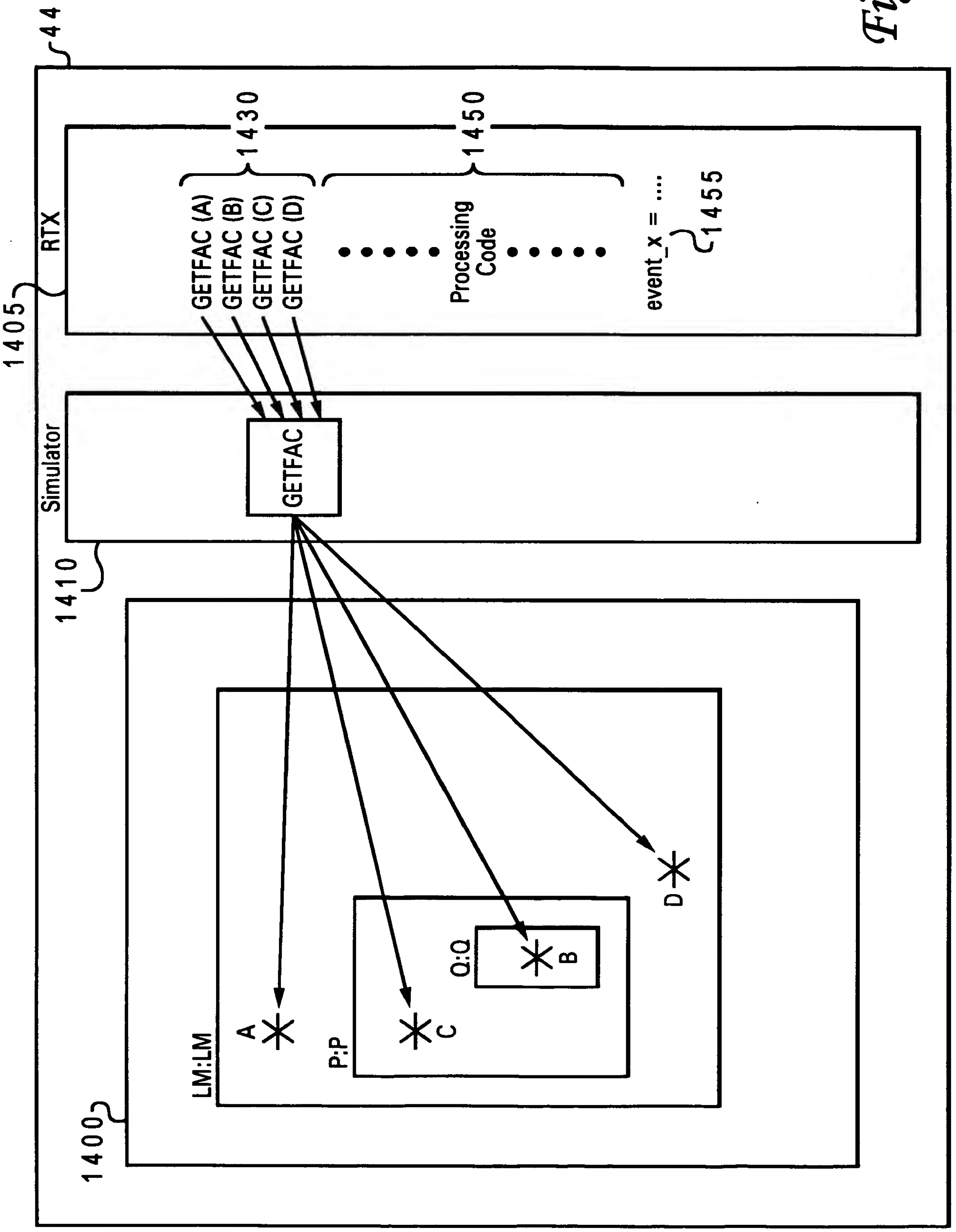

44
1405
RTX
1430
1450
GETFAC (A)
GETFAC (B)
GETFAC (C)
GETFAC (D)
Processing Code
event_x = ...
1455
Simulator
GETFAC
1410
1400
LM:LM
P:P
Q:Q
A
B
C
D

Fig. 14A

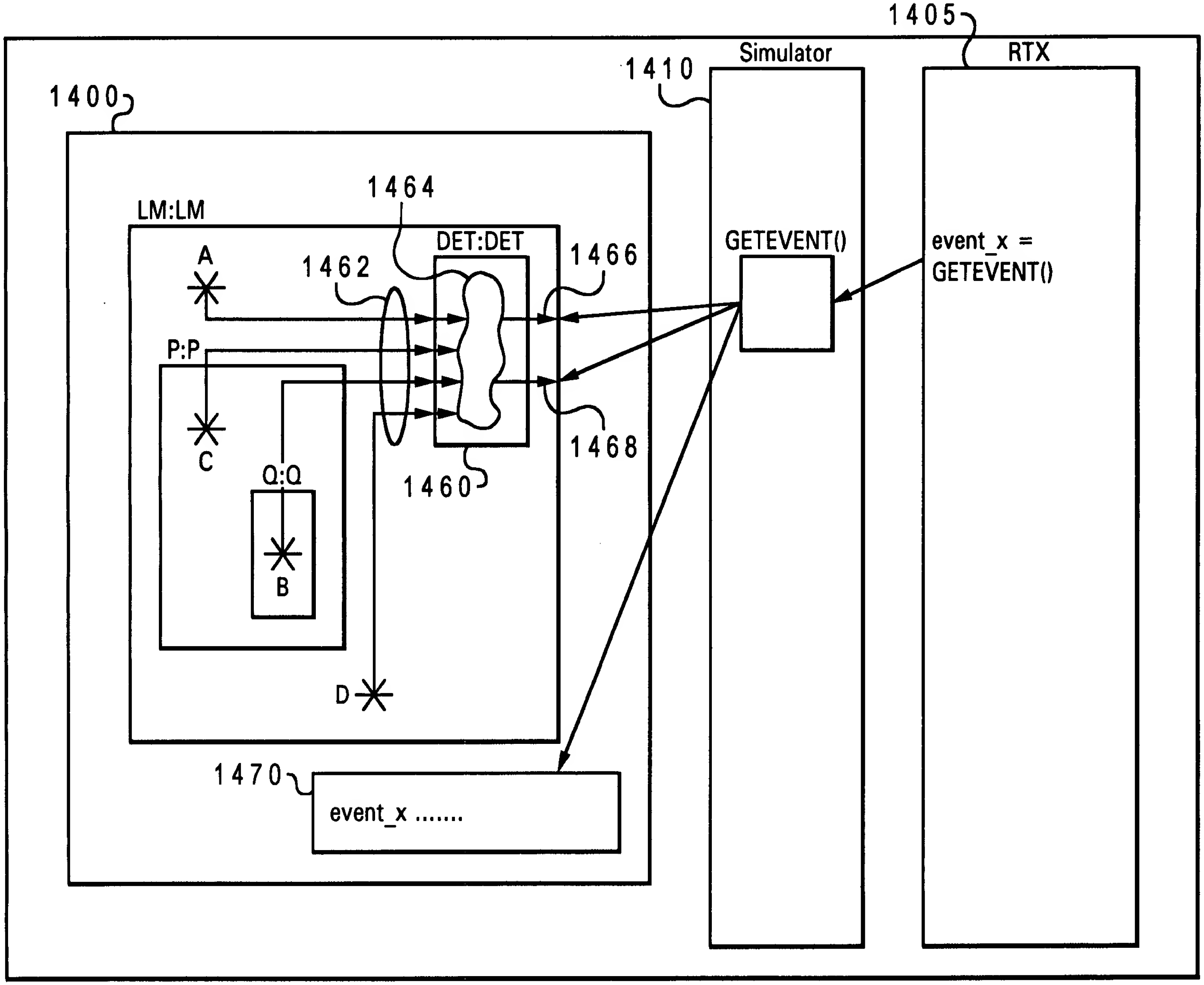
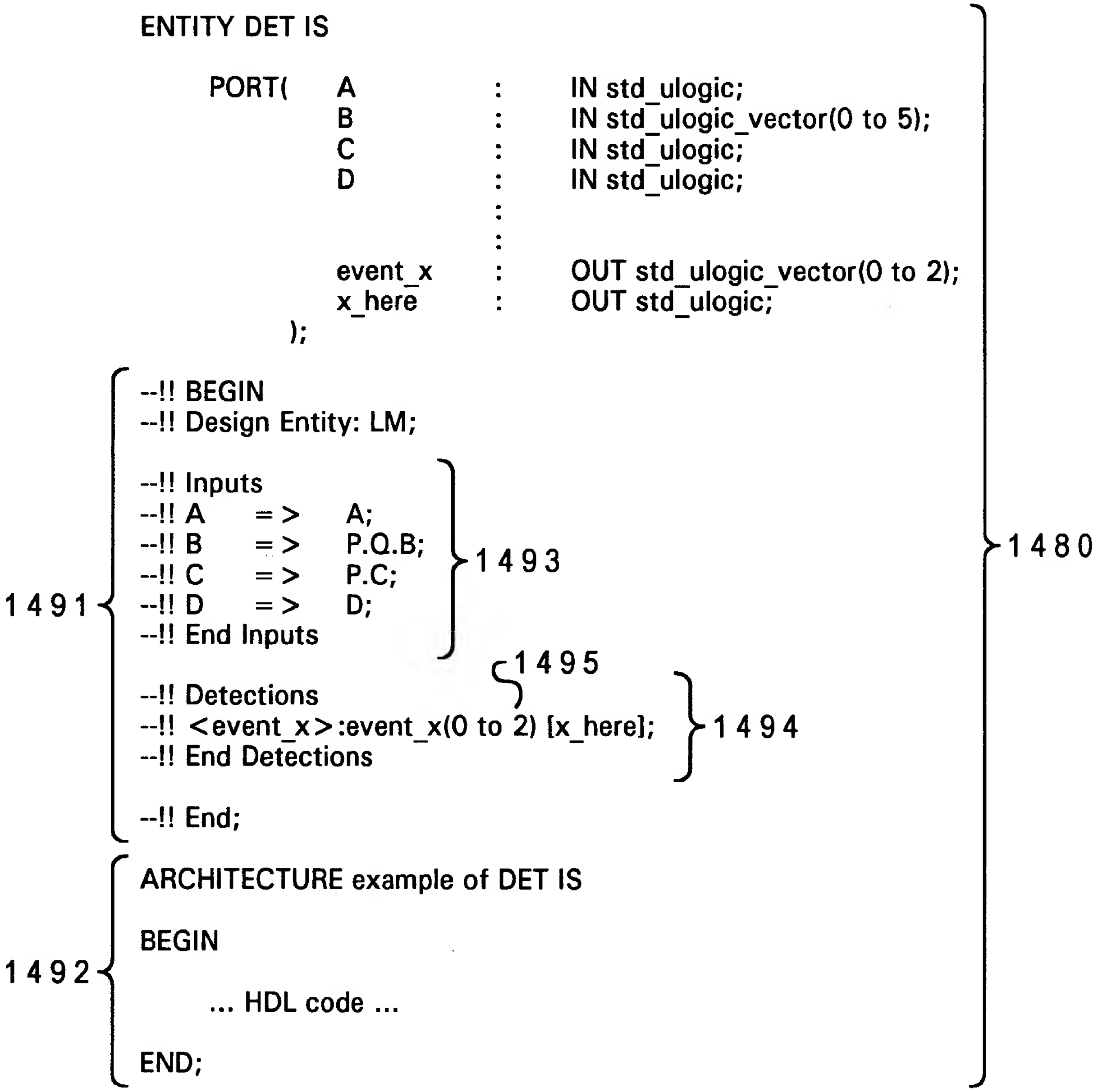

1405
1410
Simulator
RTX
1400
LM:LM
1464
DET:DET
1466
1462
A
P:P
C
Q:Q
B
1460
1468
D
GETEVENT()
event_x =
GETEVENT()
1470
event_x ....

Fig. 14B



```
ENTITY DET IS

      PORT(    A          :      IN std_ulogic;
               B          :      IN std_ulogic_vector(0 to 5);
               C          :      IN std_ulogic;
               D          :      IN std_ulogic;
                          :
                          :
               event_x    :      OUT std_ulogic_vector(0 to 2);
               x_here     :      OUT std_ulogic;
           );

--!! BEGIN
--!! Design Entity: LM;

--!! Inputs
--!! A      =>     A;
--!! B      =>     P.Q.B;
--!! C      =>     P.C;
--!! D      =>     D;
--!! End Inputs

--!! Detections
--!! <event_x>:event_x(0 to 2) [x_here];
--!! End Detections

--!! End;

ARCHITECTURE example of DET IS

BEGIN

      ... HDL code ...

END;
```

Fig. 14C